SharkFest'25 EUROPE

A Wireshark-driven approach to

MPLS

Juan Pablo Azar Ricciardi - Pierre Besombes

#sf25eu

Pierre & Juan – Who are we?

- Network Engineers @ Rakuten
  - Dealing with network infrastructures (Backbone and DCs) and some other things
  - Part of a build and operation team,
  - Some previous experiences in the SP world and large enterprises,

https://imgflip.com/memegenerator

Agenda

The goal of this talk is to

- Offer a Wireshark-driven approach to MPLS,
- Articulate the control plane and data plane's inner workings through packet analysis.
- Examine packet structures, label exchange mechanisms
- Discuss some capture and troubleshooting aspects

Sorry, but we won't talk much about RSVP-TE, SR-TE, MVPN, L2VPN… :-)

Introduction

Why MPLS in 2025?

- Why do we talk about MPLS in 2025
  - SD-WAN, SASE, AIOPS, [whatever_buzzword]…,
  - MPLS is a core/foundational network technology (mainly in the service provider and enterprise world),
  - Many large networks rely on it,
  - There are recent evolutions with SR-MPLS…,

Let's shine some light on this technology with our favorite packet analysis tool!

What is MPLS?

- What can you tell about this?

```
25 70.631720801  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27 70.635712365  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29 70.639597350  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31 70.643780263  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)
```

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
```

What is MPLS?

- What can you tell about this?

```
25 70.631720801  1.1.1.1    7.7.7.1    ICMP    118 Echo (ping) request  id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27 70.635712365  1.1.1.1    7.7.7.1    ICMP    118 Echo (ping) request  id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29 70.639597350  1.1.1.1    7.7.7.1    ICMP    118 Echo (ping) request  id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31 70.643780263  1.1.1.1    7.7.7.1    ICMP    118 Echo (ping) request  id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)
```

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
```

EtherType 0x8847

What is MPLS?

- What can you tell about this?

25	70.631720801	1.1.1.1	7.7.7.1	ICMP	118 Echo (ping) request id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27	70.635712365	1.1.1.1	7.7.7.1	ICMP	118 Echo (ping) request id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29	70.639597350	1.1.1.1	7.7.7.1	ICMP	118 Echo (ping) request id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31	70.643780263	1.1.1.1	7.7.7.1	ICMP	118 Echo (ping) request id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
```

EtherType 0x8847

Type: MPLS label switched packet (0x8847)

What is MPLS?

- What can you tell about this?

```
25 70.631720801  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27 70.635712365  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29 70.639597350  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31 70.643780263  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)
```

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
MultiProtocol Label Switching Header, Label: 16007, Exp: 0, S: 1, TTL: 255
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 7.7.7.1
Internet Control Message Protocol
```

What is MPLS?

```
25 70.631720801  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27 70.635712365  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29 70.639597350  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31 70.643780263  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)
```

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
MultiProtocol Label Switching Header, Label: 16007, Exp: 0, S: 1, TTL: 255
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 7.7.7.1
Internet Control Message Protocol
```

```
Frame 23: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
MultiProtocol Label Switching Header, Label: 16007, Exp: 0, S: 1, TTL: 255
  0000 0011 1110 1000 0111 .... .... .... = MPLS Label: 16007
  .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
  .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 7.7.7.1
Internet Control Message Protocol
```

What is MPLS?

```
25 70.631720801  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=1/256, ttl=255 (reply in 26)
27 70.635712365  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=2/512, ttl=255 (reply in 28)
29 70.639597350  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=3/768, ttl=255 (reply in 30)
31 70.643780263  1.1.1.1   7.7.7.1   ICMP   118 Echo (ping) request  id=0xe0ce, seq=4/1024, ttl=255 (reply in 32)
```

```
Frame 31: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
MultiProtocol Label Switching Header, Label: 16007, Exp: 0, S: 1, TTL: 255
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 7.7.7.1
Internet Control Message Protocol
```

```
Frame 23: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface eth0, id 0
Ethernet II, Src: 50:04:00:01:00:04 (50:04:00:01:00:04), Dst: 50:04:00:03:00:02 (50:04:00:03:00:02)
MultiProtocol Label Switching Header, Label: 16007, Exp: 0, S: 1, TTL: 255
  0000 0011 1110 1000 0111 .... .... .... = MPLS Label: 16007
  .... .... .... .... .... 000. .... .... = MPLS Experimental Bits: 0
  .... .... .... .... .... ...1 .... .... = MPLS Bottom Of Label Stack: 1
  .... .... .... .... .... .... 1111 1111 = MPLS TTL: 255
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 7.7.7.1
Internet Control Message Protocol
```

MPLS header

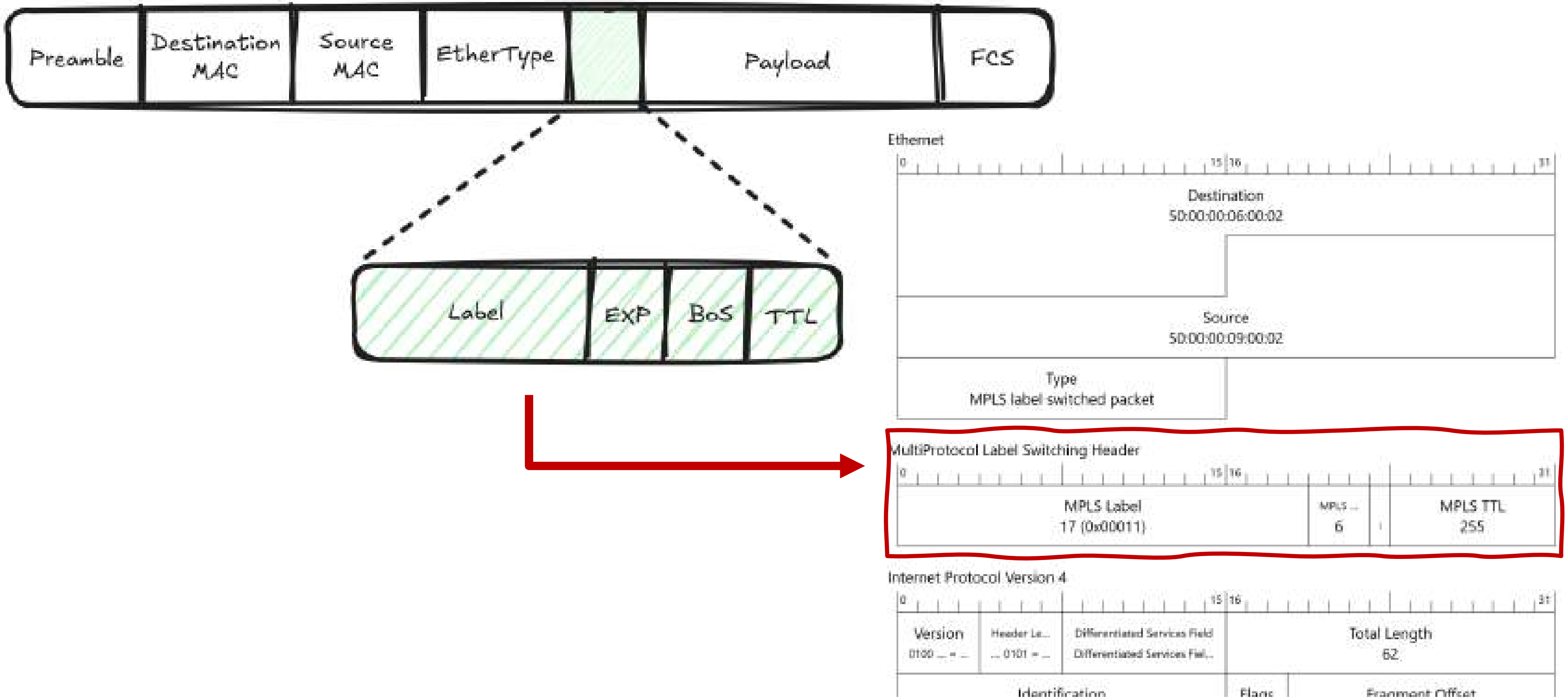
```
mpls
eth.type == 0x8847 || eth.type == 0x8848
eth.type in { 0x8847, 0x8848 }
mpls.label
```

What is MPLS?

- MPLS (Multi-Protocol Label Switching)
  - Technology relying on “labels” to perform forwarding decision
  - Label is inserted using a shim-header after Ethernet header
  - Initially used to speed up forwarding lookups (better performance using labels vs IP routing), now it is used to give you flexibility and make it possible to run various types of services (abstraction)
  - Fields: (Total 32 bits = 4 bytes)
    - Label (20 bits): *numeric value of the label,*
    - Experimental bits (3bits): *field that is leverage for QoS usually, also called TC,*
    - Bottom of stack (1bit): *when multiple labels are present is false (0), indicates the last label when true (1)*
    - Time to live TTL (8 bits): *prevents loop prevention, decremented by each router performing label switching along the path.*

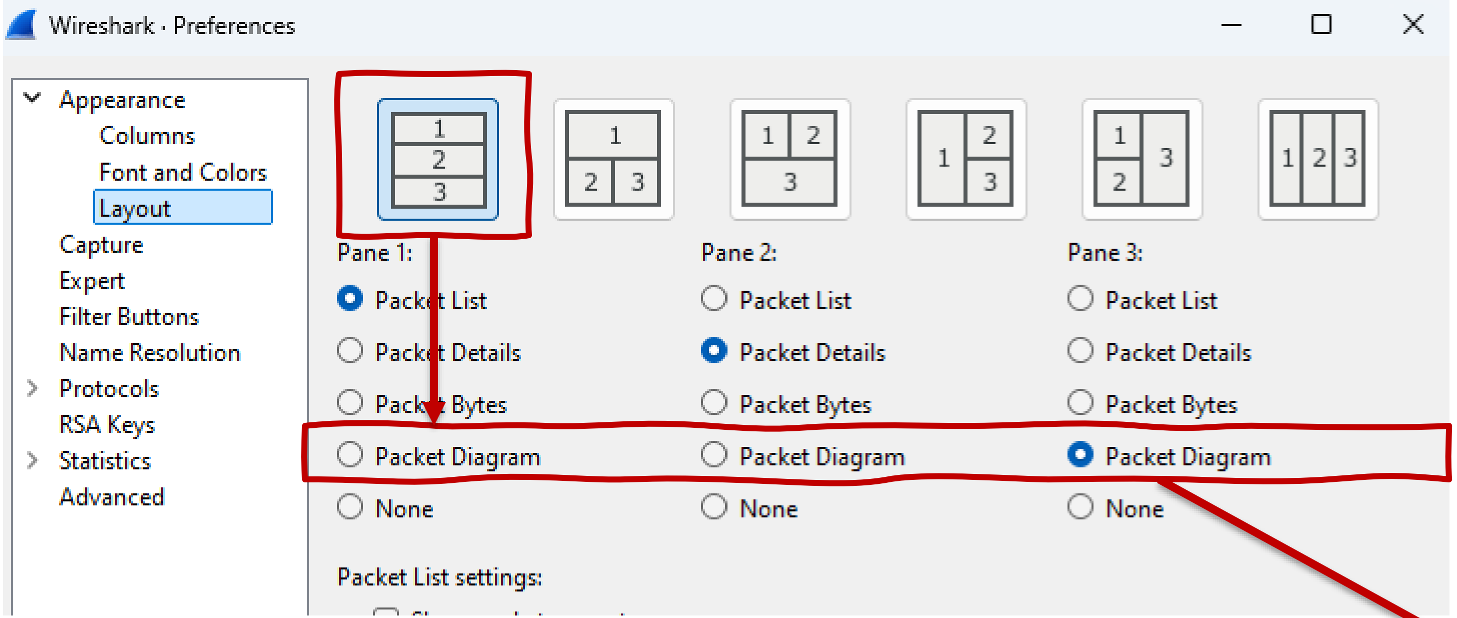
Packet Diagram

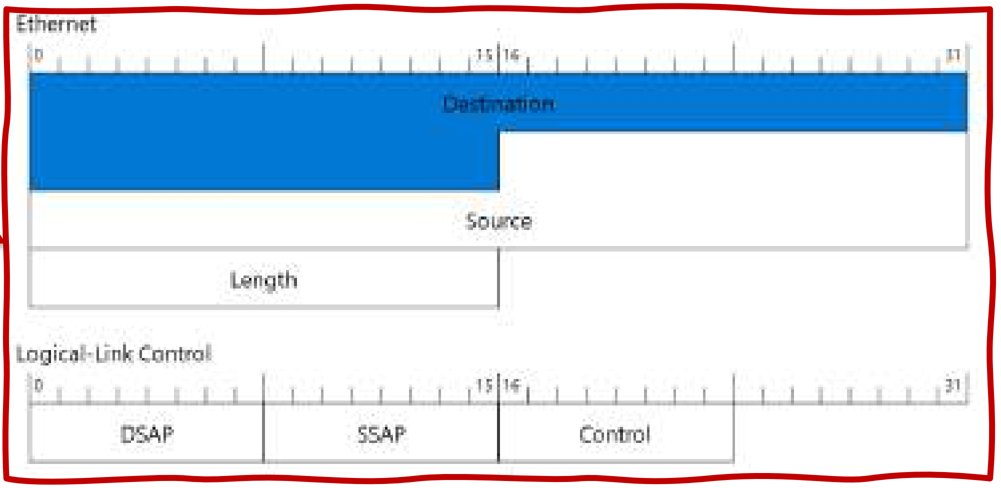
- Frame with MPLS header



Packet Diagram

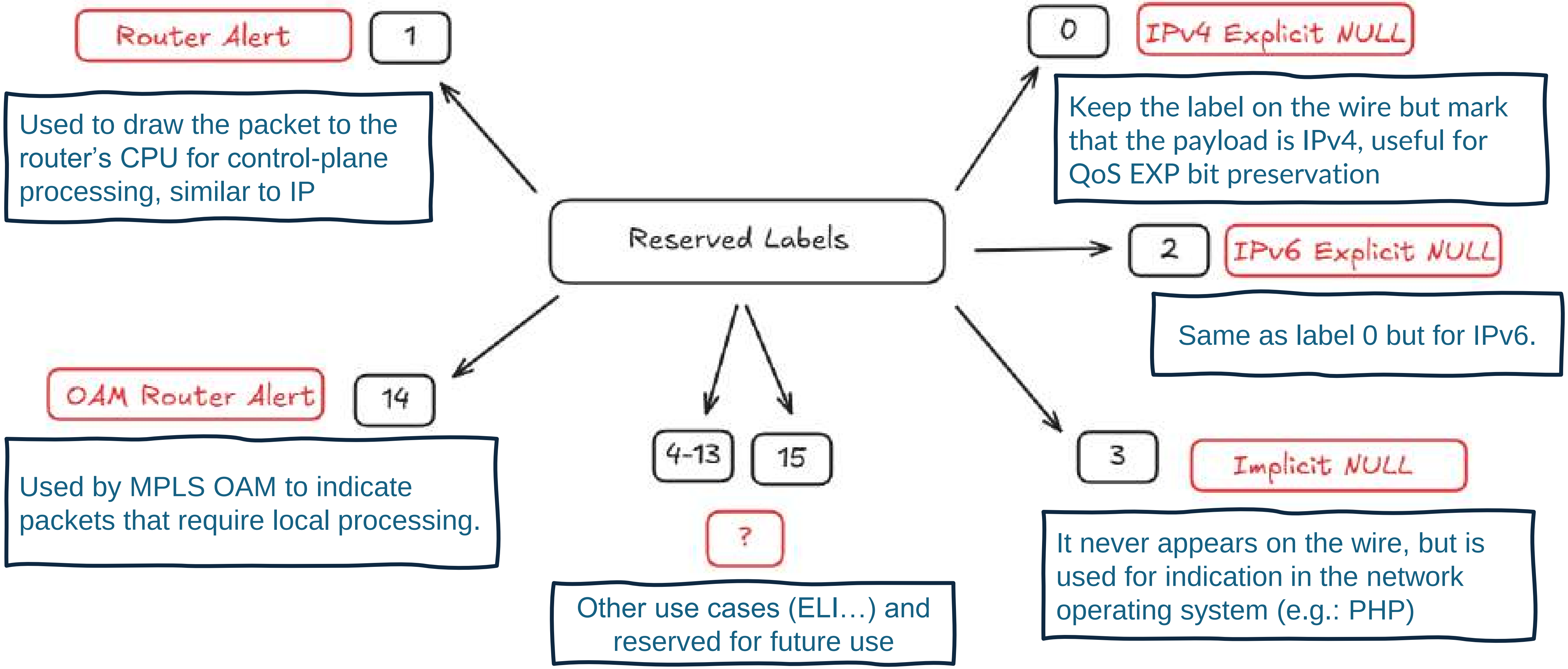
- Wireshark packet diagram for learning/visualization purposes
  - Under Preferences > layout





  - And you can also save the packet diagram as an image

Special Labels



MPLS lingo: navigating the acronyms

- Let's get through some terminology
  - CE, PE, P (Customer Equipment, Provider Edge, Provider) routers
    - Routers located at the customer site, the provider edge and inside the backbone, respectively
  - LSP (Label Switched Path)
    - The path a packet takes through an MPLS network, defined by a sequence of labels.
  - LER (Label Edge Router)
    - **Push** the first label (ingress LER) and **pop** the last label (egress LER).
  - LSR (Label Switch Router)
    - Inside the backbone, **swaps** labels based on forwarding tables. Never looks inside the IP packet header (or others headers)

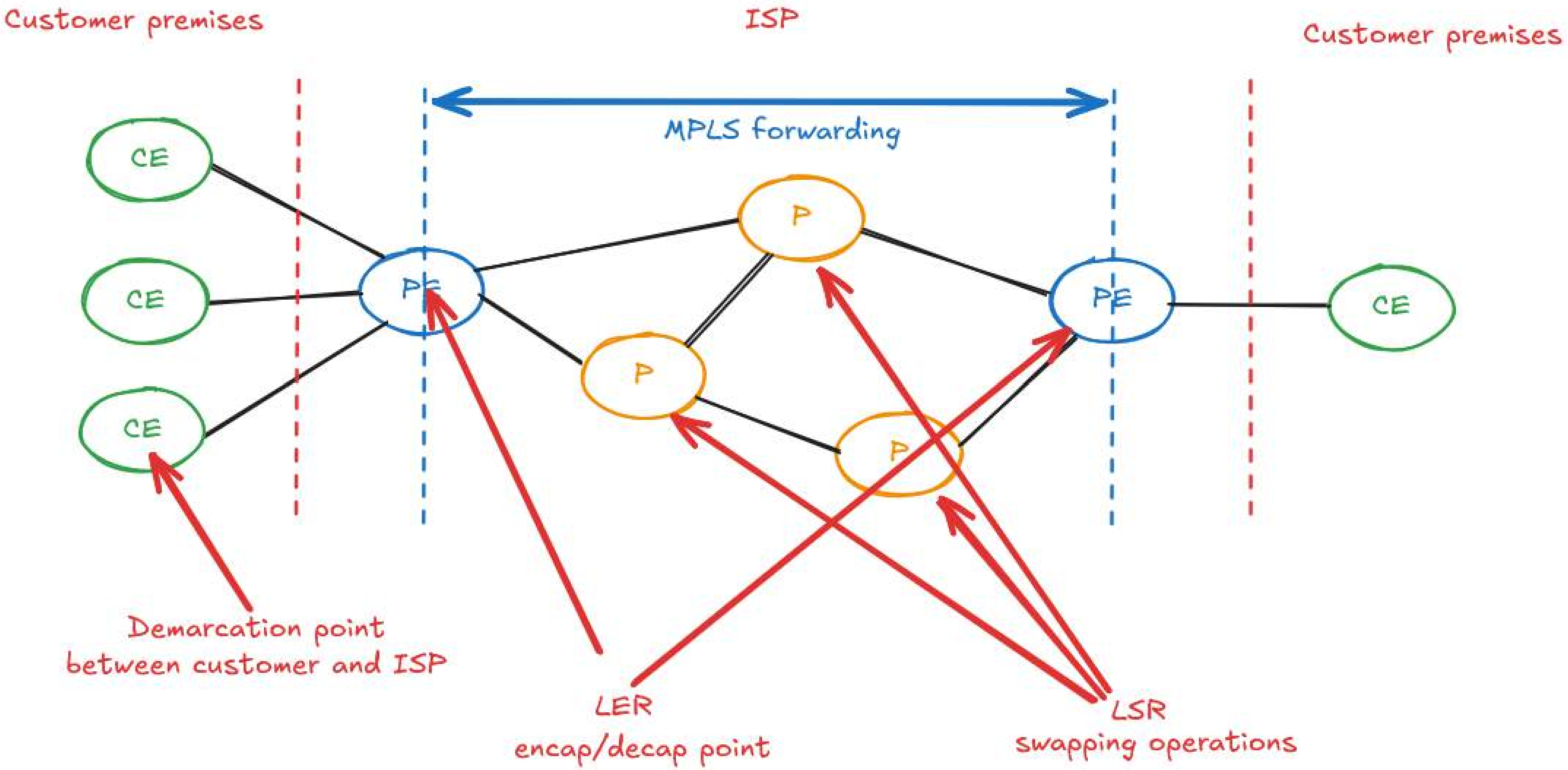
MPLS lingo: navigating the acronyms

- Let's get through some terminology
  - CE, PE, P (Customer Equipment, Provider Edge, Provider) routers
    - Routers located at the customer site, the provider edge and inside the backbone, respectively
  - LSP (Label Switched Path)
    - The path a packet takes through an MPLS network, defined by a sequence of labels.
  - LER (Label Edge Router)
    - **Push** the first label (ingress LER) and **pop** the last label (egress LER).
  - LSR (Label Switch Router)

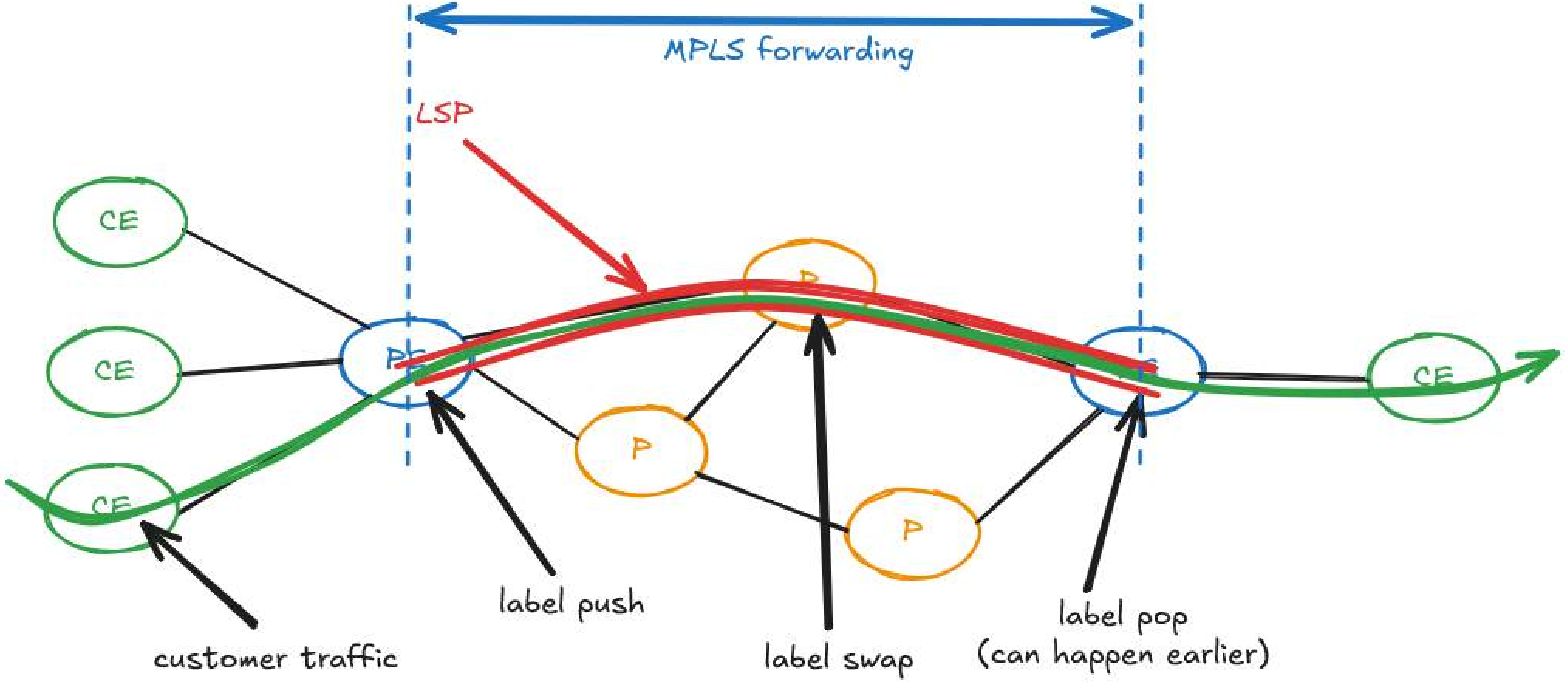
Pablo's note here, usually we have :

- **LSR = P**
- **LER = PE**
- **Why different names for same thing … because we can ?**

MPLS lingo: navigating the acronyms



MPLS lingo: navigating the acronyms



MPLS lingo cont'd

- Let's get through some terminology
  - **Signaling:** the process by which MPLS speakers (LERs, LSRs) communicate and agree on which labels to use for which FECs, and how to build the LSPs end-to-end,
  - **FEC (Forwarding Equivalence Class):** a way to treat the packet. Packets that share the same characteristics will be assigned a class and will follow the same path (a LSP in our case)
  - **MPLS VPN**: create a separated virtual network space (layer 3 or layer 2) over the shared provider infrastructure,
  - **PHP (Penultimate Hop Popping):** The second-to-last LSR that pops the top label

MPLS tables and forwarding

- Forwarding decisions
  - Routers build different forwarding tables to deal with a label-based forwarding paradigm
  - LIB: Label Information Base
  - LFIB: Label Forwarding Information Base
  - FIB: Forwarding Information Base

The LIB contains all the known label mappings on a router. It stores locally generated labels as well as those learned from neighbors by signaling protocols

The LFIB is the actual MPLS forwarding table used by the router. It maps incoming labels to outgoing labels, outgoing interfaces, next-hop IP addresses

The FIB is the actual IP forwarding table datastructure. It contains network and outgoing interfaces to quickly forward traffic

MPLS tables and forwarding

SharkFest'25 EUROPE

LIB

```
RP/0/RP0/CPU0:P3#show mpls ldp bindings
Tue Oct 28 13:36:37.002 UTC

1.1.1.1/32, rev 28
        Local binding: label: 24008
        Remote bindings: (3 peers)
            Peer                    Label
            -----------------       ---------
            3.3.3.3:0               16
            11.11.11.11:0           24008
            22.22.22.22:0           24009
2.2.2.2/32, rev 32
        Local binding: label: 24010
        Remote bindings: (3 peers)
            Peer                    Label
            -----------------       ---------
            3.3.3.3:0               17
            11.11.11.11:0           24010
            22.22.22.22:0           24011
```

FIB

```
RP/0/RP0/CPU0:P3#show ip cef
Tue Oct 28 13:44:24.412 UTC

Prefix              Next Hop            Interface
------------------- ------------------- ------------------
0.0.0.0/0           drop                default handler
0.0.0.0/32          broadcast
1.1.1.1/32          10.11.33.11/32      GigabitEthernet0/0/0/0
2.2.2.2/32          10.22.33.22/32      GigabitEthernet0/0/0/2
3.3.3.3/32          10.33.3.3/32        GigabitEthernet0/0/0/3
4.4.4.4/32          10.33.44.44/32      GigabitEthernet0/0/0/1
10.1.11.0/24        10.11.33.11/32      GigabitEthernet0/0/0/0
10.2.22.0/24        10.22.33.22/32      GigabitEthernet0/0/0/2
10.11.22.0/24       10.11.33.11/32      GigabitEthernet0/0/0/0
                    10.22.33.22/32      GigabitEthernet0/0/0/2
10.11.33.0/24       attached            GigabitEthernet0/0/0/0
10.11.33.0/32       broadcast           GigabitEth
```

LFIB

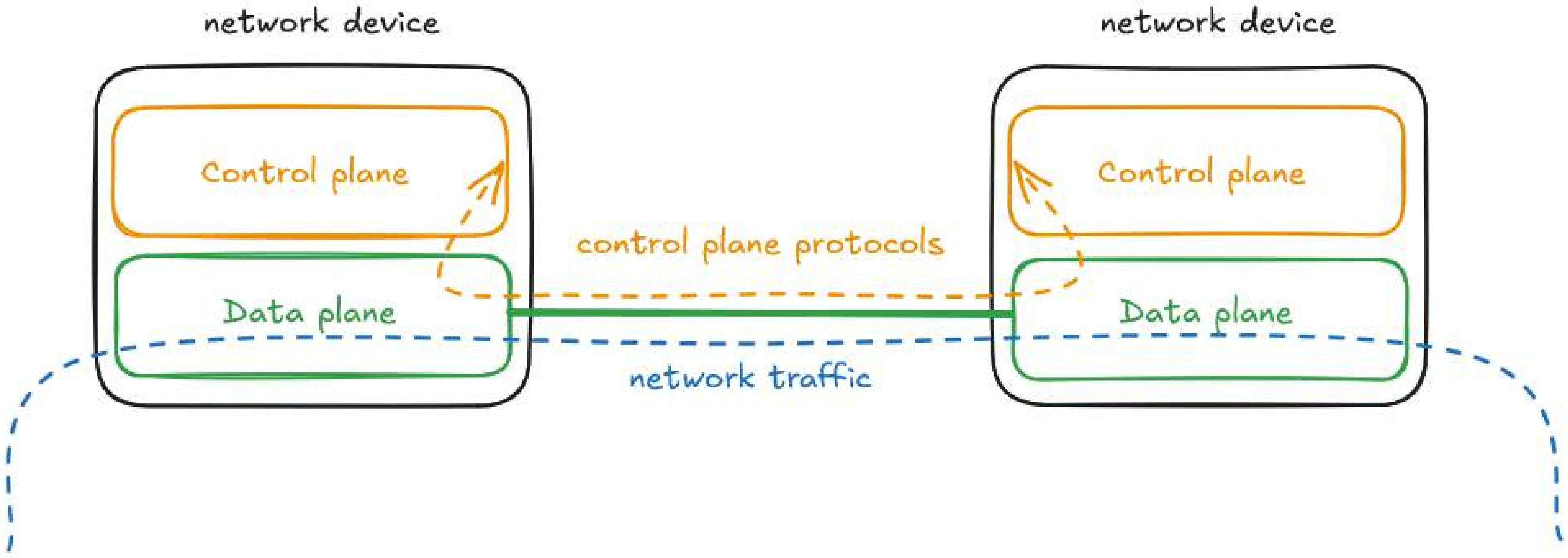
```
RP/0/RP0/CPU0:P3#show mpls forwarding
Tue Oct 28 13:41:07.284 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------------
24000  Pop         11.11.11.11/32     Gi0/0/0/0    10.11.33.11     526267
24001  Unlabelled  44.44.44.44/32     Gi0/0/0/1    10.33.44.44     0
24002  Pop         10.11.44.0/24      Gi0/0/0/0    10.11.33.11     0
       Unlabelled  10.11.44.0/24      Gi0/0/0/1    10.33.44.44     0
24003  Pop         10.1.11.0/24       Gi0/0/0/0    10.11.33.11     0
24004  Pop         10.11.22.0/24      Gi0/0/0/0    10.11.33.11     0
       Pop         10.11.22.0/24      Gi0/0/0/2    10.22.33.22     0
24005  Unlabelled  10.22.44.0/24      Gi0/0/0/1    10.33.44.44     0
24006  Unlabelled  10.44.4.0/24       Gi0/0/0/1    10.33.44.44     0
24007  Pop         22.22.22.22/32     Gi0/0/0/2    10.22.33.22     526772
24008  24008       1.1.1.1/32         Gi0/0/0/0    10.11.33.11     0
24009  Pop         10.2.22.0/24       Gi0/0/0/2    10.22.33.22     0
24010  24011       2.2.2.2/32         Gi0/0/0/2    10.22.33.22     0
24011  Pop         3.3.3.3/32         Gi0/0/0/3    10.33.3.3       617720
24012  Unlabelled  4.4.4.4/32         Gi0/0/0/1    10.33.44.44     0
```

Pablo's note here :

- The tables for real (Cisco IOS-XR)

Control and data plane

- How to build these tables?
  - Using some network control plane protocols
- The control plane is the brain of the device, computing and pushing the forwarding information to the data plane.

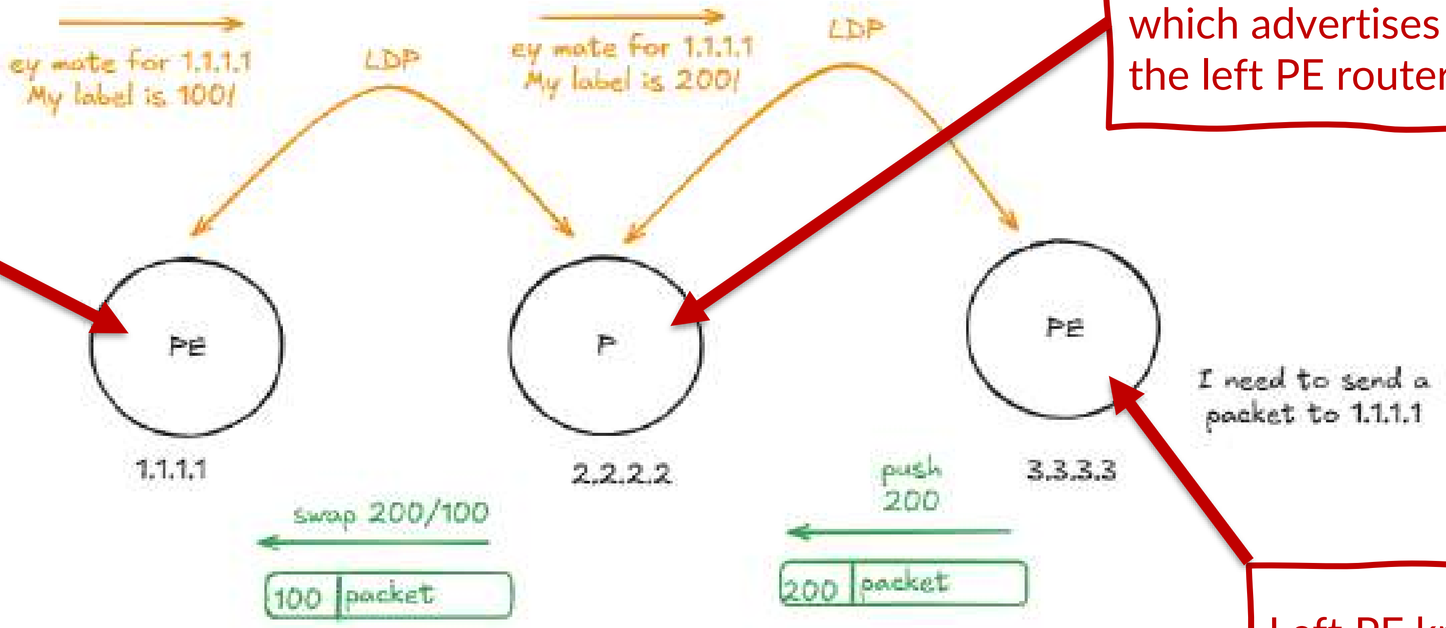


To make it clear !

- To show an easy example of transport labels

In this case, the PE advertises (via LDP) that if the P router needs to forward a labeled packet towards 1.1.1.1, it should use label 100.

The same happens with the P router, which advertises label 200 towards the left PE router



Left PE knows which label to use to send to 1.1.1.1.

* *Since this is a simple example, we are skipping labels 0 and 3, as well as PHP behavior for now.*

Remember those tables? Let's see them in action!

We can see how the local label assignment determines how our neighbors will handle the labeled packet they receive – whether they will push, swap, or pop the 'Implicit Null' corresponds to label 3, one of the reserved labels, we've seen that earlier!

PE

```
PE1#show mpls ldp bindings
  lib entry: 1.1.1.1/32, rev 28
        local binding:  label: imp-null
        remote binding: lsr: 11.11.11.11:0, label: 24008
        remote binding: lsr: 3.3.3.3:0, label: 16
```

P

```
1.1.1.1/32, rev 28
        Local binding: label: 24008
        Remote bindings: (3 peers)
            Peer                Label
            -----------------   ---------
            10.1.11.1:0         ImpNull
            22.22.22.22:0       24009
            33.33.33.33:0       24008
```

```
RP/0/RP0/CPU0:ios#show mpls forwarding prefix 1.1.1.1/32
Mon Nov  3 09:38:14.042 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop
Label  Label       or ID              Interface
------ ----------- ------------------ ------------ ----------
24008  Pop         1.1.1.1/32         Gi0/0/0/3    10.1.11.1
```

PE

```
Mon Nov  3 09:43:08.967 UTC
1.1.1.1/32, rev 28
        Local binding: label: 24008
        Remote bindings: (3 peers)
            Peer                Label
            -----------------   --------
            3.3.3.3:0           16
            11.11.11.11:0       24008
            44.44.44.44:0       24008
```

```
Local  Outgoing    Prefix             Outgoing     Next Hop
Label  Label       or ID              Interface
------ ----------- ------------------ ------------ ----------
24008  24008       1.1.1.1/32         Gi0/0/0/0    10.11.33.11
```

Label distribution

- So, to impose labels on IP packets, routers need to have label tables populated.
  - Various protocols can be used to exchange labels and do LSP "signaling", as well as defining labels for specific resources (a VPN for example), this, depending on the use case
    - LDP (Label Discovery Protocol)
    - RSVP-TE (Resource Reservation Protocol – Traffic Engineering)
    - MP-BGP with VPNv4 (L3VPN), IPv4-LU (Labelled Unicast), Ethernet VPN (L2VPN)
    - IS-IS/OSPF (used in Segment Routing, to derive labels)
    - Without a protocol, statically (use case: configured through a controller)

Label distribution

SharkFest'25 EUROPE

- So, to impose labels on IP packets, routers need to have label tables populated.
  - Various protocols can be used to exchange labels and do LSP "signaling", as well as defining labels for specific resources (a VPN for example), this, depending on the use case
    - LDP (Label Discovery Protocol)
    - RSVP-TE (Resource Reservation Protocol – Traffic Engineering)
    - MP-BGP with VPNv4 (L3VPN), IPv4-LU (Labelled Unicast), Ethernet VPN (L2VPN)
    - IS-IS/OSPF (used in Segment Routing, to derive labels)
    - Without a protocol, statically (use case: configured through a controller)

Traffic Engineering

Internal label exchange

Label exchange between networks

Advanced use cases

Services (L*VPNs, PW…)

Capture

- What can you tell about this?

59.693550324	3.3.3.3	646	4.4.4.4	276…	TCP
59.703382359	3.3.3.3	646	4.4.4.4	276…	TCP
59.705051118	4.4.4.4	646	3.3.3.3	646	UDP
59.714887541	4.4.4.4	27625	3.3.3.3	646	TCP
59.726286564	3.3.3.3	646	4.4.4.4	276…	TCP
59.927527777	4.4.4.4	27625	3.3.3.3	646	TCP
61.912422350	4.4.4.4	646	2.2.2.2	646	UDP
62.100330854	20.2.4.4	646	224.0.0.2	646	UDP
62.635315519	20.2.4.2	646	224.0.0.2	646	UDP
64.986977563	2.2.2.2	36055	6.6.6.6	179	TCP
66.147517129	4.4.4.4	646	2.2.2.2	646	UDP
66.500335926	20.2.4.2	646	224.0.0.2	646	UDP
66.837254000	20.2.4.4	646	224.0.0.2	646	UDP
70.028042398	4.4.4.4	646	2.2.2.2	646	UDP
70.860110277	20.2.4.2	646	224.0.0.2	646	UDP
71.418084747	20.2.4.4	646	224.0.0.2	646	UDP

Capture

- What can you tell about this?

59.693550324	3.3.3.3	646	4.4.4.4	276…	TCP
59.703382359	3.3.3.3	646	4.4.4.4	276…	TCP
59.705051118	4.4.4.4	646	3.3.3.3	646	UDP
59.714887541	4.4.4.4	27625	3.3.3.3	646	TCP
59.726286564	3.3.3.3	646	4.4.4.4	276…	TCP
59.927527777	4.4.4.4	27625	3.3.3.3	646	TCP
61.912422350	4.4.4.4	646	2.2.2.2	646	UDP
62.100330854	20.2.4.4	646	224.0.0.2	646	UDP
62.635315519	20.2.4.2	646	224.0.0.2	646	UDP
64.986977563	2.2.2.2	36055	6.6.6.6	179	TCP
66.147517129	4.4.4.4	646	2.2.2.2	646	UDP
66.500335926	20.2.4.2	646	224.0.0.2	646	UDP
66.837254000	20.2.4.4	646	224.0.0.2	646	UDP
70.028042398	4.4.4.4	646	2.2.2.2	646	UDP
70.860110277	20.2.4.2	646	224.0.0.2	646	UDP
71.418084747	20.2.4.4	646	224.0.0.2	646	UDP

Filter:
ldp

Capture

- LDP (TCP)

```
> Frame 134: Packet, 396 bytes on wire (3168 bits), 396 bytes captured (3168 bits) on interface eth0, id 0
> Ethernet II, Src: 50:00:00:07:00:02 (50:00:00:07:00:02), Dst: 50:00:00:05:00:02 (50:00:00:05:00:02)
> MultiProtocol Label Switching Header, Label: 16, Exp: 6, S: 1, TTL: 255
> Internet Protocol Version 4, Src: 4.4.4.4, Dst: 3.3.3.3
> Transmission Control Protocol, Src Port: 27625, Dst Port: 646, Seq: 65, Ack: 73, Len: 338
> Label Distribution Protocol
v Label Distribution Protocol
    Version: 1
    PDU Length: 316
    LSR ID: 4.4.4.4
    Label Space ID: 0
  > Address Message
  > Label Mapping Message
  > Label Mapping Message
  > Label Mapping Message
  > Label Mapping Message
```

Capture

- LDP (UDP)

```
> Frame 96: Packet, 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface eth0, id 0
> Ethernet II, Src: 50:00:00:05:00:02 (50:00:00:05:00:02), Dst: IPv4mcast_02 (01:00:5e:00:00:02)
> Internet Protocol Version 4, Src: 20.2.4.2, Dst: 224.0.0.2
> User Datagram Protocol, Src Port: 646, Dst Port: 646
v Label Distribution Protocol
    Version: 1
    PDU Length: 30
    LSR ID: 2.2.2.2
    Label Space ID: 0
  > Hello Message
```

Label Discovery Protocol (LDP)

- LDP (Label Discovery Protocol) is used in many network to perform label exchange
  - Exchange labels bindings for IP addresses
  - Rely on both TCP and UDP, port 646
    - UDP for Hello message (to 224.0.0.2, or directed)
    - TCP for reliable label exchange
  - LDP IPv4 Transport Address: is the IP used to establish the TCP session between peers (usually the loopback address).
  - Another specific case: LDP targeted sessions (e.g.: L2VPN/VPLS) established between non-direct peers (vs directed connected neighbors)
  - Different operation modes possible to be conservative or not (DoD, DU)

Label Discovery Protocol (LDP)

- LDP,
  - behaves as locally significant between two directly connected peers
  - creates LSP following the underlying IGP (IGP metrics matters here)
  - will need to be configured consistently with the IGP, this synchronization is important (or might lead to blackhole),
  - Upon IGP convergence, new paths will be reflected automatically by LDP
  - Static and directly connected routes are considered “IGP” from LDP standpoint
  - Common TE technique is to tweak IGP path costs

Capture

- What can you tell about this?

```
131 135.079235393 3.3.3.3    1.1.1.1    TCP    62 16562 → 179 [SYN] Seq=0 Win=16384 Len=0 MSS=1396
124 129.004398118 4.4.4.4    3.3.3.3    TCP    54 179 → 14845 [ACK] Seq=39 Ack=58 Win=15919 Len=0
123 128.799600246 3.3.3.3    4.4.4.4    BGP    77 KEEPALIVE Message
119 125.618780189 1.1.1.1    3.3.3.3    TCP    54 179 → 29129 [ACK] Seq=40 Ack=59 Win=16061 Len=0
118 125.615172485 3.3.3.3    1.1.1.1    TCP    58 29129 → 179 [FIN, PSH, ACK] Seq=58 Ack=40 Win=15852 Len=0
117 125.614408993 3.3.3.3    1.1.1.1    TCP    58 29129 → 179 [ACK] Seq=58 Ack=40 Win=15852 Len=0
116 125.613577384 1.1.1.1    3.3.3.3    TCP    54 179 → 29129 [FIN, PSH, ACK] Seq=39 Ack=58 Win=16061 Len=0
```

```
Frame 144: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface eth0, id 0
Ethernet II, Src: 50:02:00:10:00:06 (50:02:00:10:00:06), Dst: 50:02:00:14:00:02 (50:02:00:14:00:02)
Internet Protocol Version 4, Src: 1.1.1.1, Dst: 3.3.3.3
Transmission Control Protocol, Src Port: 179, Dst Port: 16562, Seq: 104, Ack: 272, Len: 141
Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 112
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 89
    Path attributes
        Path Attribute - MP_REACH_NLRI
            Flags: 0x80, Optional, Non-transitive, Complete
            Type Code: MP_REACH_NLRI (14)
            Length: 48
            Address family identifier (AFI): IPv4 (1)
            Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
            Next hop:  RD=0:0 IPv4=1.1.1.1
            Number of Subnetwork points of attachment (SNPA): 0
            Network Layer Reachability Information (NLRI)
                BGP Prefix
                BGP Prefix
                    Prefix Length: 120
                    Label Stack: 33 (bottom)
                    Route Distinguisher: 1:1
                    MP Reach NLRI IPv4 prefix: 192.168.100.100
        Path Attribute - ORIGIN: INCOMPLETE
        Path Attribute - AS_PATH: 65000
```

BGP VPNv4

```
117 125.614408993 3.3.3.3      1.1.1.1      TCP      58 29129 → 179 [ACK] Seq=58 Ack=40 Win=15852 Len=0
116 125.613577384 1.1.1.1      3.3.3.3      TCP      54 179 → 29129 [FIN, PSH, ACK] Seq=39 Ack=58 Win=16061 Len=0

▸ Frame 144: 195 bytes on wire (1560 bits), 195 bytes captured (1560 bits) on interface eth0, id 0
▸ Ethernet II, Src: 50:02:00:10:00:06 (50:02:00:10:00:06), Dst: 50:02:00:14:00:02 (50:02:00:14:00:02)
▸ Internet Protocol Version 4, Src: 1.1.1.1, Dst: 3.3.3.3
▾ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 112
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
  ▾ Path attributes
    ▾ Path Attribute - MP_REACH_NLRI
      ▸ Flags: 0x80, Optional, Non-transitive, Complete
        Type Code: MP_REACH_NLRI (14)
        Length: 48
        Address family identifier (AFI): IPv4 (1)
        Subsequent address family identifier (SAFI): Labeled VPN Unicast (128)
      ▸ Next hop:  RD=0:0 IPv4=1.1.1.1
        Number of Subnetwork points of attachment (SNPA): 0
      ▾ Network Layer Reachability Information (NLRI)
        ▸ BGP Prefix
        ▾ BGP Prefix
            Prefix Length: 120
            Label Stack: 33 (bottom)
            Route Distinguisher: 1:1
            MP Reach NLRI IPv4 prefix: 192.168.100.100
    ▸ Path Attribute - ORIGIN: INCOMPLETE
    ▸ Path Attribute - AS_PATH: 65000
```

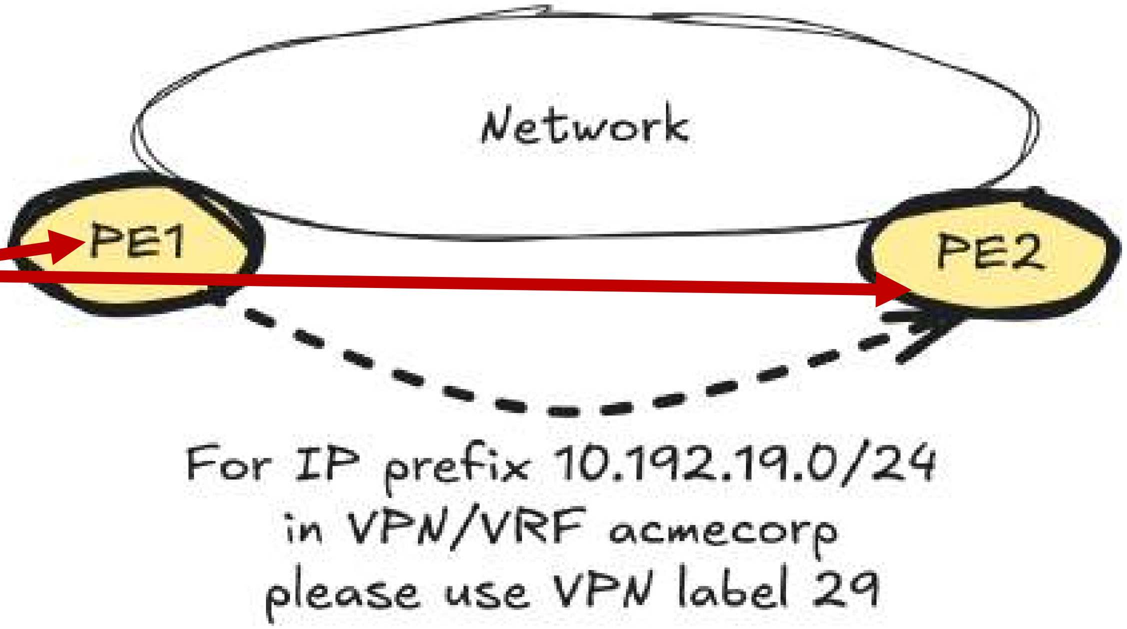
```
PE3#show bgp vpnv4 unicast all neighbors 1.1.1.1 routes
BGP table version is 17, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path, L long-lived-stale,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:1
 *>i  172.16.101.0/24  1.1.1.1                  0    100      0 65000 ?
 *>i  192.168.100.100/32
                       1.1.1.1                  0    100      0 65000 ?
```

BGP VPNv4 (VPNv6)

- BGP can be used to exchange labels in specific scenarios
  - **BGP is a TCP application**, using well know port 179
  - BGP is an extensible protocol that can carry multiple **address families**
    - We'll focus on VPNv4 address family, which will signal the label associated with a VRF (routing/forwarding table instance, aka VPN)
    - Each VRF is separate from the others

Goal is to distribute the label associated to a prefix and propagate it elsewhere in or out a network. This label being received on a remote ingress LERs (PEs) indicate it what to impose to reach this specific destination



Pablo's Note:
- The VPN label assigned by the PE is not necessarily the same for all prefixes within the same VPN.
- For example, PE1 might assign label 30 for prefix x.x.x.x/x and label 31 for prefix y.y.y.y/y — both in the same VRF.
- It could be per-prefix or per-table.

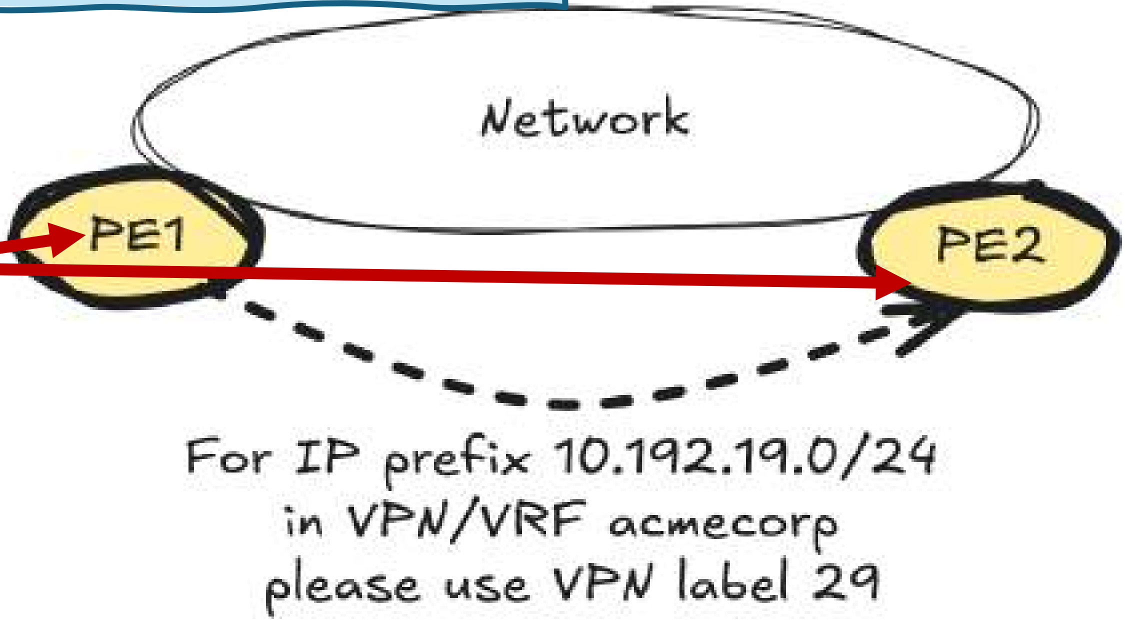
c scenarios

9

ble **address families**

label associated with a

Goal is to distribute the label associated to a prefix and propagate it elsewhere in or out a network. This label being received on a remote ingress LERs (PEs) indicate it what to impose to reach this specific destination



BGP - filtering BGP messages

SharkFest'25 EUROPE

You can quickly isolate BGP messages using the message type

https://datatracker.ietf.org/doc/html/rfc4271

```
Type: UPDATE Message (2)
```

1 - OPEN

`bgp.type == 1`

When you look for BGP peering initialization parameters

2 - UPDATE

`bgp.type == 2`

When you want to see routing updates

3 - NOTIFICATION

`bgp.type == 3`

Notifications are error messages. Can be useful when troubleshooting

4 - KEEPALIVE

`!bgp.type == 4`

Filter out KEEPALIVE, good to do when you have aggressive timers, like in datacenters.

`bgp.type in {1,2,3}`

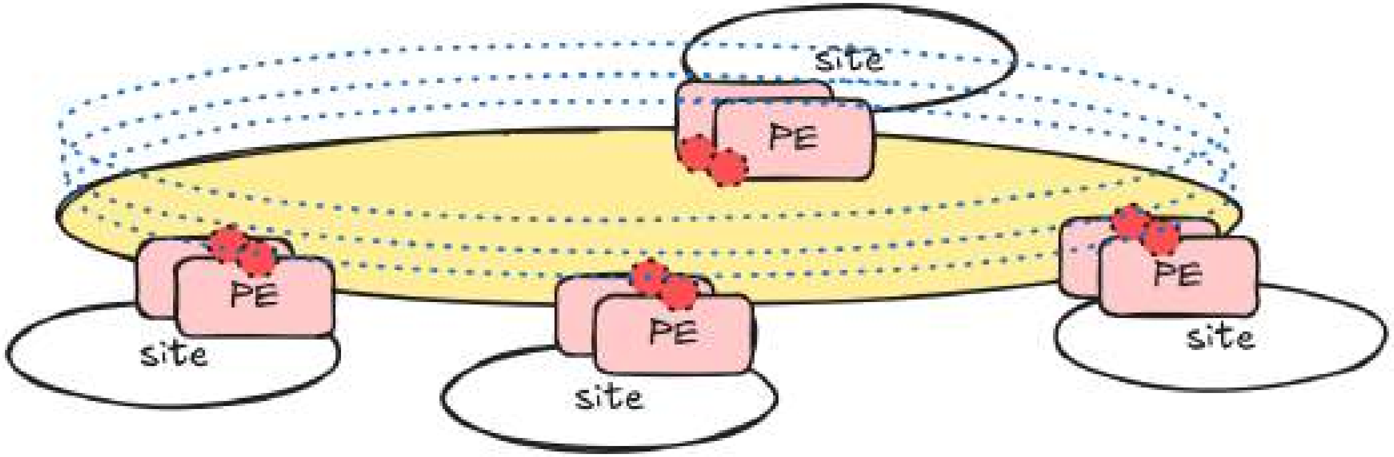
MP-BGP - AFI/SAFIMP

- BGP AFI/SAFI structure is defined under RFC4760 and maintained by IANA
  - https://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml
  - https://www.iana.org/assignments/safi-namespace/safi-namespace.xhtml
- Each assignment is also described in the related protocol extension RFC

83	BGP CAR	[RFC-ietf-idr-bgp-car-16]
84	BGP VPN CAR	[RFC-ietf-idr-bgp-car-16]
85	BGP-MUP SAFI	[draft-mpmz-bess-mup-safi-00]
86-127	Unassigned	
128	MPLS-labeled VPN address	[RFC4364][RFC8277][RFC9252]
129	Multicast for BGP/MPLS IP Virtual Private Networks (VPNs)	[RFC6513][RFC6514]
130-131	Reserved	[RFC4760]
132	Route Target constrains	[RFC4684]
133	Dissemination of Flow Specification rules	[RFC8955]
134	L3VPN Dissemination of Flow Specification rules	[RFC8955]

MPLS VPNs

- MPLS VPNs (Virtual Private Network)
  - Allows service providers to carry multiple customer networks over a shared backbone, transparently,
  - VPN service separation is achieved
    - On control-plane side, with VRF, Route Distinguishers, and Route Targets, labels
    - On data-plane side, through label switching that provide an abstraction layer,
  - VPNs can be layer 2 or layer 3, signaled differently
  - MPLS core remains labels switched (BGP free-core, label forwarding only)



MPLS VPNs

- A few more pieces,
  - **Route distinguishers**
    - Unique identifiers to make IP prefixes unique and distinguish them in an L3VPN environment (the IP prefix becomes a VPN prefix)
    - Ensure that identical IP prefixes can coexist inside different VRF/VPNs,
    - 64-bit value, often in the format of AS:NN or IP:NN, where AS is the autonomous system number or IP is an IP address (IP of the PE), and NN is a unique number.
  - **BGP Communities**
    - are used to tag routes with values that can be later matched and used to influence routing decisions and policies (two types, standard and extended communities)
  - **Route targets**
    - Route targets are extended BGP communities used to control the import and export of routes in L3VPNs.
    - Often in the format of AS:NN (0x00) or IP:NN (0x01),

MPLS VPNs

- MPLS VPNs (Virtual Private Network)
  - VPNs communication is based on at least two labels, a VPN label and one or more transport label(s)
  - VPN labels:
    - **L3VPN:** exchanged with MP-BGP with VPNv4/VPNv6 routes and their associated labels for unicast, multicast traffic can also be carried (MVPN, various flavors)
    - **L2VPN:** exchanged with MP-BGP EVPN / VPLS or LDP, different options there too for signaling

This top label will be swapped along the path by P routers, the service one will remain intact (exchanged between PE routers with MP-BGP).

Capture

- Capture, between P routers

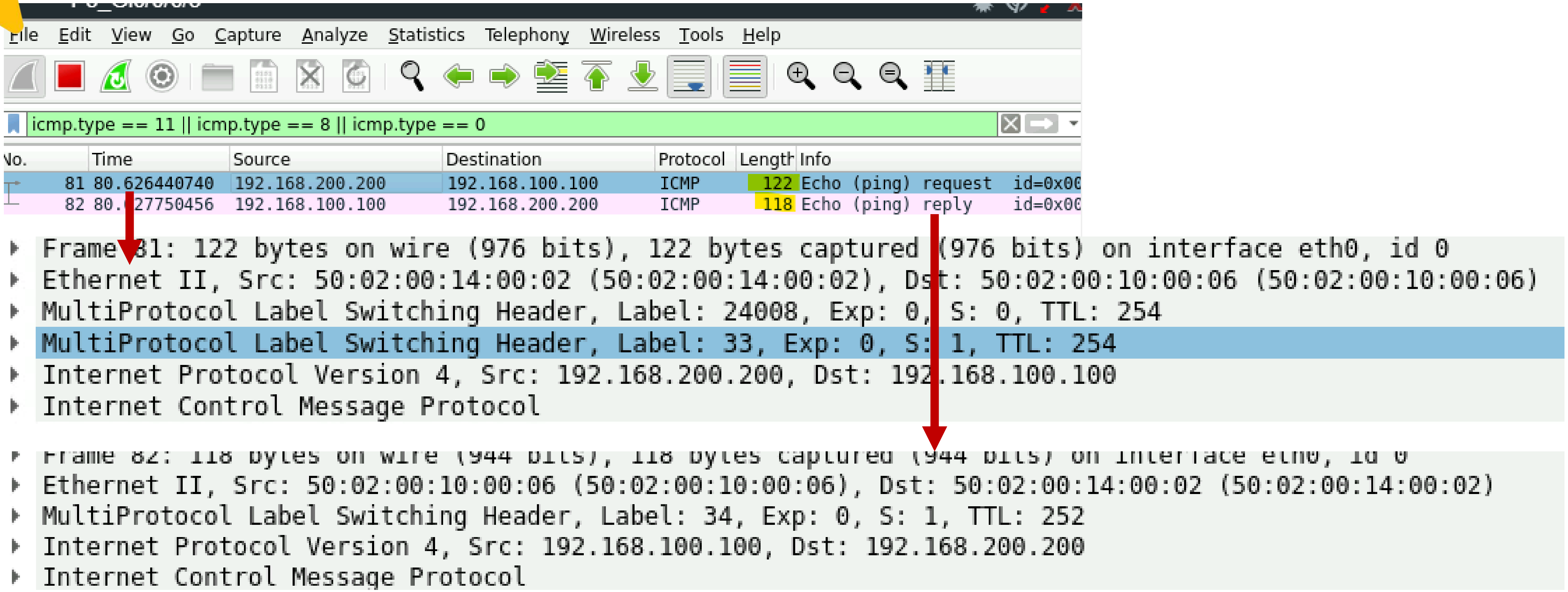
No.	Time	Source	Destination	Protocol	Length	Info
89	105.095003583	192.168.200.200	192.168.100.100	ICMP	122	Echo (ping) request id=0x00
90	105.095880477	192.168.100.100	192.168.200.200	ICMP	122	Echo (ping) reply id=0x00

```
Frame 89: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface eth0, id 0
Ethernet II, Src: 50:02:00:10:00:03 (50:02:00:10:00:03), Dst: 50:02:00:0f:00:03 (50:02:00:0f:00:03)
MultiProtocol Label Switching Header, Label: 24008, Exp: 0, S: 0, TTL: 253
MultiProtocol Label Switching Header, Label: 33, Exp: 0, S: 1, TTL: 254
Internet Protocol Version 4, Src: 192.168.200.200, Dst: 192.168.100.100
Internet Control Message Protocol
```

```
Frame 90: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface eth0, id 0
Ethernet II, Src: 50:02:00:0f:00:03 (50:02:00:0f:00:03), Dst: 50:02:00:10:00:03 (50:02:00:10:00:03)
MultiProtocol Label Switching Header, Label: 24011, Exp: 0, S: 0, TTL: 253
MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
Internet Control Message Protocol
```

Capture

- Traffic is reaching the destination. What happens between last P – PE segment on the return path?



Capture

- Finally, PE-CE capture

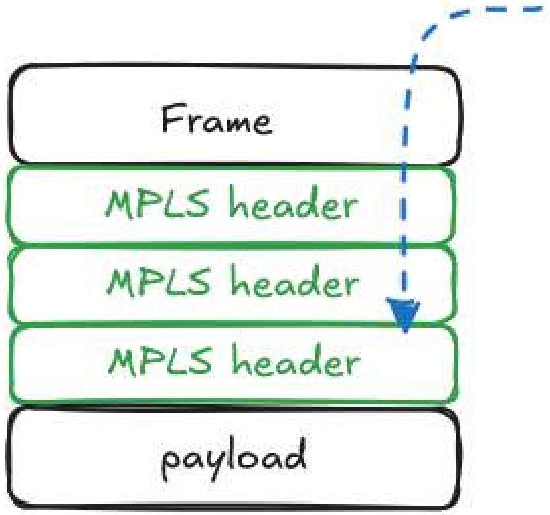
No.	Time	Source	Destination	Protocol	Length	Info
5	40.695845819	192.168.200.200	192.168.100.100	ICMP	114	Echo (ping) request id=0x
6	40.697472590	192.168.100.100	192.168.200.200	ICMP	114	Echo (ping) reply id=0x

No more labels

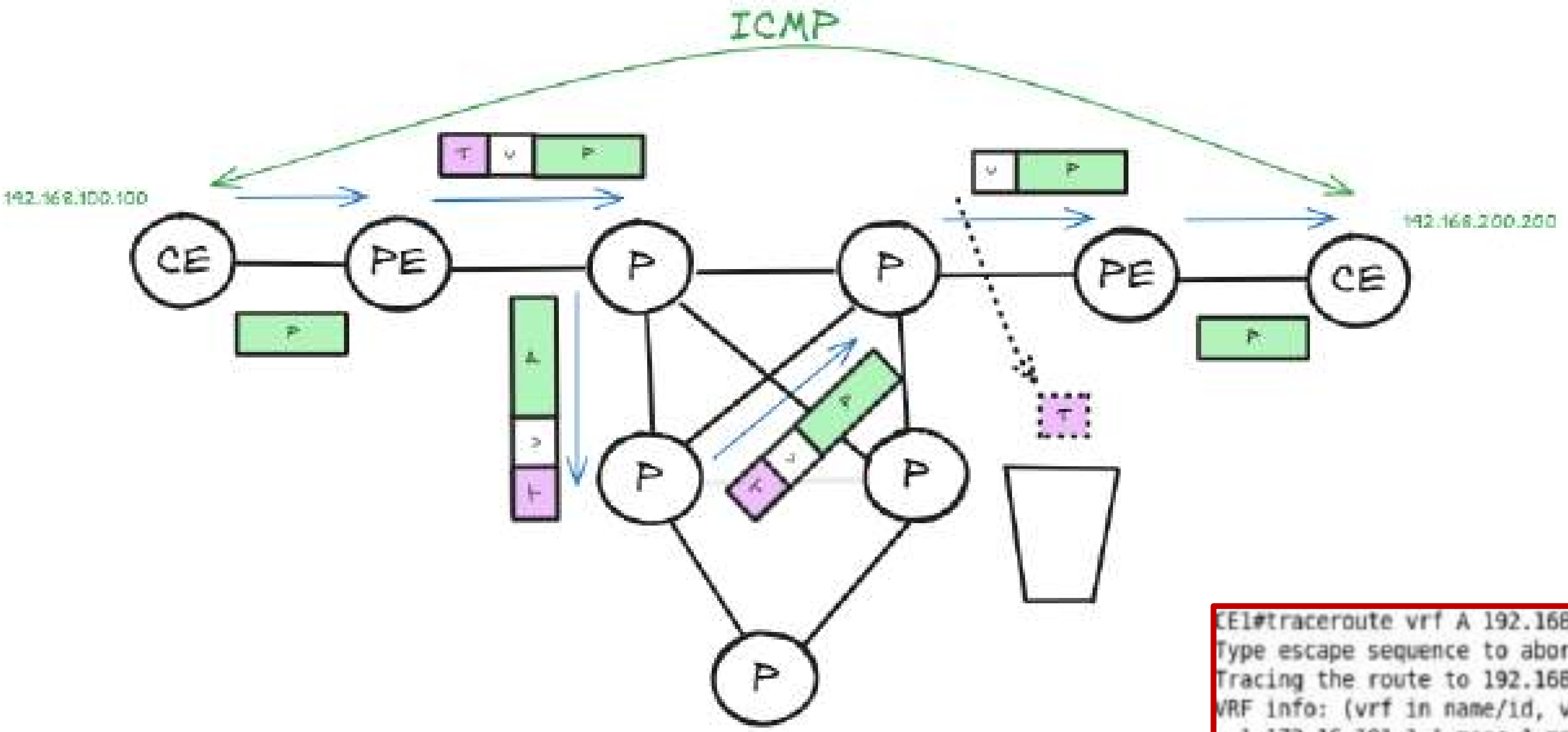
```
Frame 5: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface eth0, id 0
Ethernet II, Src: 50:02:00:18:00:00 (50:02:00:18:00:00), Dst: 50:02:00:14:00:00 (50:02:00:14:00:00)
Internet Protocol Version 4, Src: 192.168.200.200, Dst: 192.168.100.100
Internet Control Message Protocol
```

Label stack

- Label stack
  - **label stack** is a series of **MPLS headers** inserted
  - Each label represents a **forwarding instruction,** LSRs use it to decide where the packet goes next, without looking into the IP header.
  - Labels are **stacked** in a **Last-In, First-Out (LIFO)** manner.
  - The **top label** (outermost) determines the current forwarding action.
  - S-bit (**Bottom of Stack**/BoS) marks which label is the last one before the payload



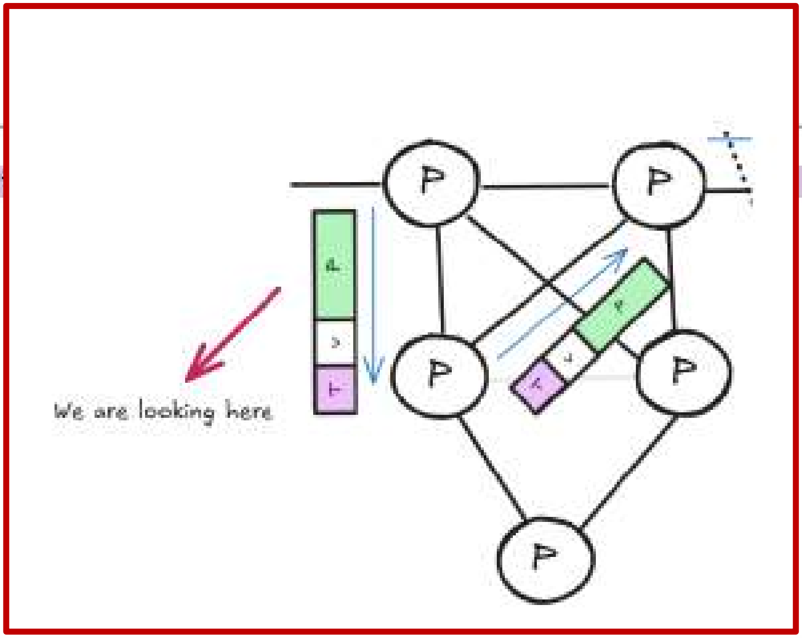
Walk Together




```
CE1#traceroute vrf A 192.168.200.200 source 192.168.100.100
Type escape sequence to abort.
Tracing the route to 192.168.200.200
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.101.1 1 msec 1 msec 0 msec
  2 10.1.11.11 [MPLS: Labels 24014/34 Exp 0] 4 msec 4 msec 2 msec
  3 10.11.22.22 [MPLS: Labels 24013/34 Exp 0] 3 msec 4 msec 2 msec
  4 10.22.33.33 [MPLS: Labels 24010/34 Exp 0] 4 msec 4 msec 2 msec
  5 172.16.203.3 [MPLS: Label 34 Exp 0] 2 msec 2 msec 2 msec
  6 172.16.203.2 3 msec 3 msec *
```

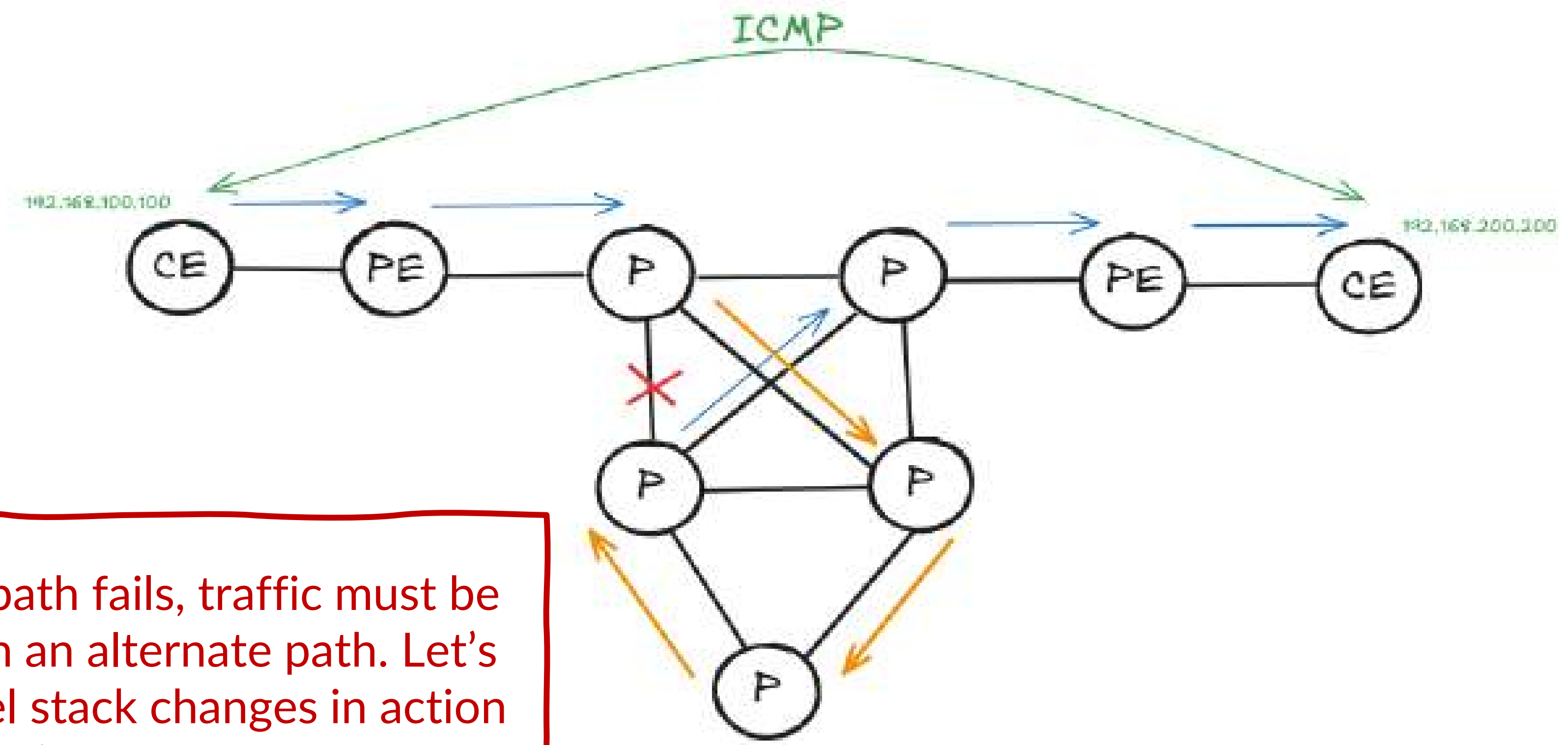
Walk Together

No.	Time	Source	Destination	Protocol	Length	Info
51	45.872990991	192.168.100.100	192.168.200.200	ICMP	122	Echo (ping) request id=0x000c, seq=0/0,



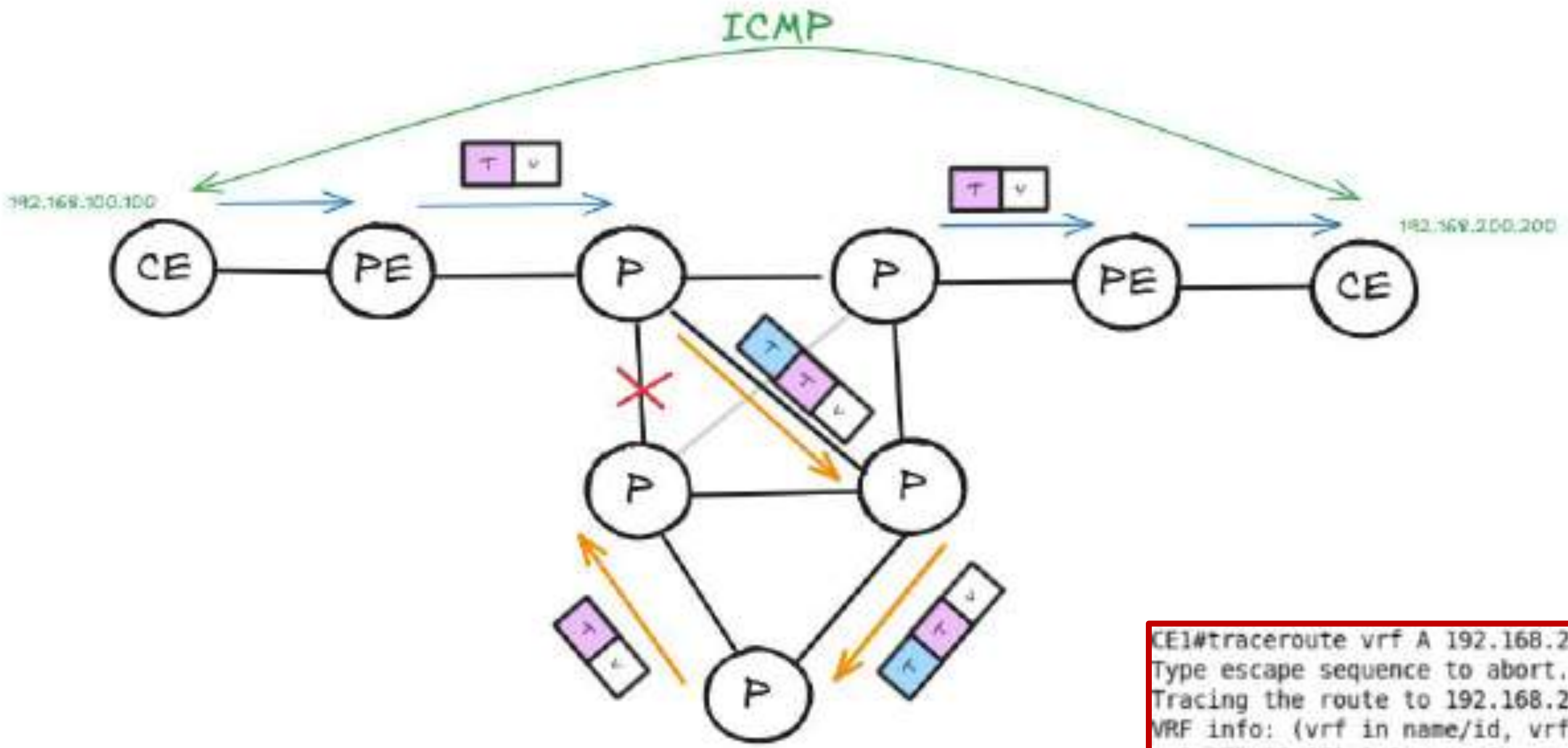
Frame 51: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
Ethernet II, Src: 50:02:00:0f:00:04 (50:02:00:0f:00:04), Dst: 50:02:00:11:00:04 (50:02:00:11:00:04)
MultiProtocol Label Switching Header, Label: 24012, Exp: 0, S: 0, TTL: 253
MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
Internet Control Message Protocol

Rerouting, label stack in action



When the main path fails, traffic must be rerouted through an alternate path. Let’s see how the label stack changes in action during link protection.

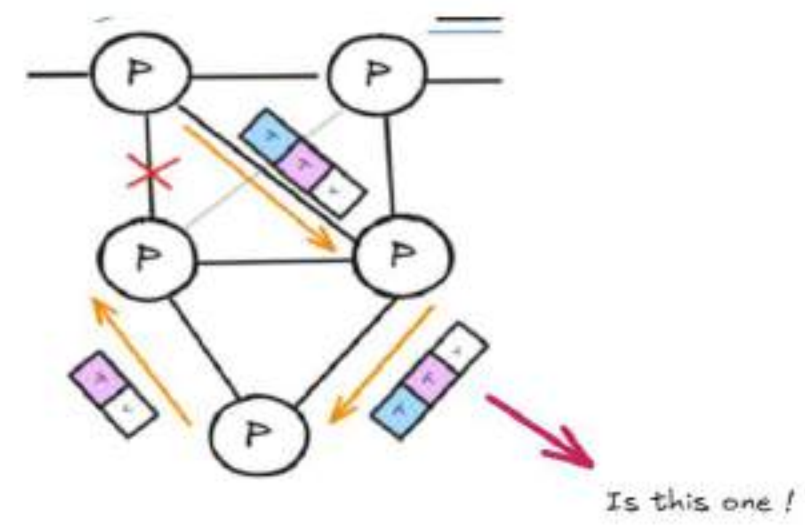
Rerouting, label stack in action




```
CE1#traceroute vrf A 192.168.200.200 source 192.168.100.100
Type escape sequence to abort.
Tracing the route to 192.168.200.200
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.101.1 1 msec 0 msec 0 msec
  2 10.1.11.11 [MPLS: Labels 24014/34 Exp 0] 4 msec 5 msec 3 msec
  3 10.11.44.44 [MPLS: Labels 24010/24013/34 Exp 0] 5 msec 4 msec 3 msec
  4 10.44.4.4 [MPLS: Labels 24/24013/34 Exp 0] 3 msec 4 msec 3 msec
  5 10.2.22.22 [MPLS: Labels 24013/34 Exp 0] 3 msec 5 msec 3 msec
  6 10.22.33.33 [MPLS: Labels 24010/34 Exp 0] 3 msec 4 msec 3 msec
  7 172.16.203.3 [MPLS: Label 34 Exp 0] 2 msec 5 msec 3 msec
  8 172.16.203.2 3 msec 4 msec *
```

Rerouting, label stack in action

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.100.100	192.168.200.200	ICMP	126	Echo (ping)

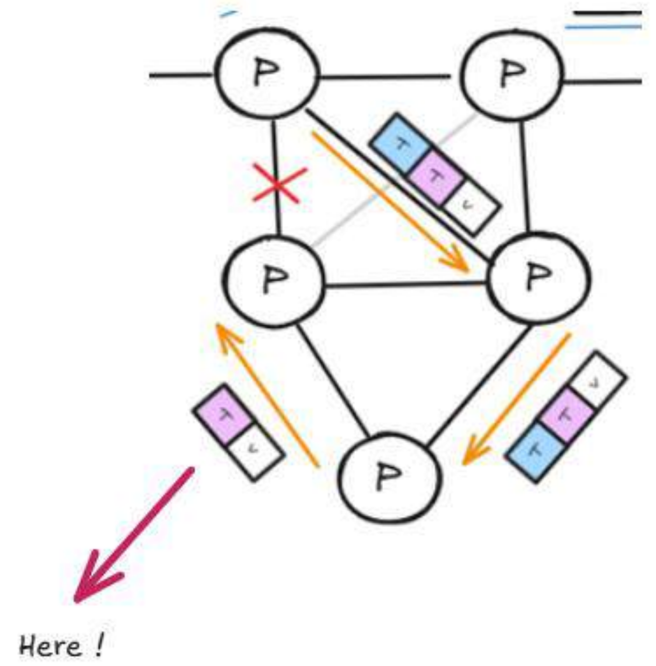



```
> Frame 1: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0
> Ethernet II, Src: 50:02:00:12:00:06 (50:02:00:12:00:06), Dst: 50:02:00:16:00:02 (50:02:00:16:00:02)
> MultiProtocol Label Switching Header, Label: 24, Exp: 0, S: 0, TTL: 252
> MultiProtocol Label Switching Header, Label: 24012, Exp: 0, S: 0, TTL: 253
> MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
> Internet Control Message Protocol
```

Rerouting, label stack in action

No.	Time	Source	Destination	Protocol	Length	Info
291	282.170863356	192.168.100.100	192.168.200.200	ICMP	122	Echo (ping) request id=0x000c

```
> Frame 291: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: 50:02:00:16:00:00 (50:02:00:16:00:00), Dst: 50:02:00:11:00:06 (50:02:00:11:00:06)
> MultiProtocol Label Switching Header, Label: 24012, Exp: 0, S: 0, TTL: 251
> MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
> Internet Control Message Protocol
```

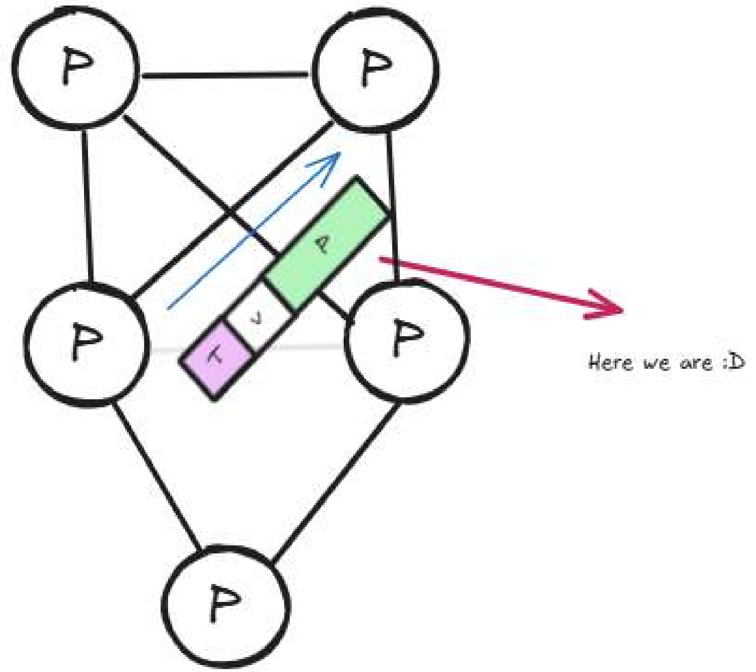


Rerouting, label stack in action

No.	Time	Source	Destination	Protocol	Length	Info
27	22.966277592	192.168.100.100	192.168.200.200	ICMP	122	Echo (ping) request id=0x000c, seq=0/0, ttl=254 (reply in 28)
110	88.280769747	192.168.100.100	192.168.200.200	ICMP	122	Echo (ping) request id=0x000d, seq=0/0, ttl=254 (reply in 111)

```
> Frame 110: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: 50:02:00:11:00:05 (50:02:00:11:00:05), Dst: 50:02:00:10:00:05 (50:02:00:10:00:05)
> MultiProtocol Label Switching Header, Label: 24005, Exp: 0, S: 0, TTL: 250
> MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
> Internet Control Message Protocol
```

```
> Frame 27: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0
> Ethernet II, Src: 50:02:00:11:00:05 (50:02:00:11:00:05), Dst: 50:02:00:10:00:05 (50:02:00:10:00:05)
> MultiProtocol Label Switching Header, Label: 24005, Exp: 0, S: 0, TTL: 252
> MultiProtocol Label Switching Header, Label: 34, Exp: 0, S: 1, TTL: 254
> Internet Protocol Version 4, Src: 192.168.100.100, Dst: 192.168.200.200
> Internet Control Message Protocol
```



Capture

- What can you tell about this?

```
69 144.175304…          50:04:00:…          DEC-MAP-(or-OSI?)-Intermediate-System-Hello?

› Frame 90: Packet, 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface eth0, id 0
˅ IEEE 802.3 Ethernet
  › Destination: DEC-MAP-(or-OSI?)-Intermediate-System-Hello? (09:00:2b:00:00:05)
  › Source: 50:04:00:03:00:06 (50:04:00:03:00:06)
    Length: 169
    [Stream index: 0]
˅ Logical-Link Control
  ˅ DSAP: ISO Network Layer (0xfe)
      1111 111. = SAP: ISO Network Layer
      .... ...0 = IG Bit: Individual
  ˅ SSAP: ISO Network Layer (0xfe)
      1111 111. = SAP: ISO Network Layer
```

Capture

- What can you tell about this?

```
69 144.175304…          50:04:00:…          DEC-MAP-(or-OSI?)-Intermediate-System-Hello?

> Frame 90: Packet, 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface eth0, id 0
v IEEE 802.3 Ethernet
  > Destination: DEC-MAP-(or-OSI?)-Intermediate-System-Hello? (09:00:2b:00:00:05)
  > Source: 50:04:00:03:00:06 (50:04:00:03:00:06)
    Length: 169
    [Stream index: 0]
v Logical-Link Control
  v DSAP: ISO Network Layer (0xfe)
      1111 111. = SAP: ISO Network Layer
      .... ...0 = IG Bit: Individual
  v SSAP: ISO Network Layer (0xfe)
      1111 111. = SAP: ISO Network Layer
```

```
v ISO 10589 ISIS Link State Protocol Data Unit
    PDU length: 166
    Remaining lifetime: 1199
    LSP-ID: 0000.0000.0010.00-00
    Sequence number: 0x00001565
    Checksum: 0x8f31 [correct]
    [Checksum Status: Good]
  > Type block(0x03): Partition Repair:0, Attached bits:0, Overload bit:0, IS type:3
  > Area address(es) (t=1, l=4)
  > Protocols supported (t=129, l=1)
  > IP Interface address(es) (t=132, l=4)
  > Extended IP Reachability (t=135, l=21)
  > Hostname (t=137, l=3)
  > Router Capability (t=242, l=24)
  > Extended IS reachability (t=22, l=68)
```

```
isis
isis.lsp
isis.lsp.rt_capable.flag_s
isis.lsp.adj_sid.flags
```

IGP is IS-IS

- IS-IS (Intermediate System to Intermediate System)
  - Link state routing protocol (similar OSPF, popular in SP)
  - Originally defined ISO/CLNS scheme,
  - Runs directly on top of layer-2, no IP header there!,
  - In a nutshell, routers flood Link State PDUs (LSPs again...)
    - They carry information as TLV (Type Length Value) structures, a flexible method to carry information in your protocol,
    - Segment Routing leverages this TLV structure to carry SID (Segment IDentifier) information
    - TLV 135 (sub-TLV 131 containing adjacency SID flags), TLV 242 Router Capability (sub-TLV containing SR support algo, SRGB), TLV 22 (sub-TLV for prefix SID advertisement)
  - Used for Segment Routing MPLS (SR-MPLS)!

Capture – Router capability SR

- Router capability

```
⌄ Router Capability (t=242, l=24)
    Type: 242
    Length: 24
    Router ID: 0x01010101
    .... ...0 = S bit: False
    .... ..0. = D bit: False
  ⌄ Segment Routing - Capability (t=2, l=9)
      1... .... = I flag: IPv4 support: True
      .0.. .... = V flag: IPv6 support: False
      Range: 8000
    ⌄ SID/Label (t=1, l=3)
        Label: 16000
  ⌄ Segment Routing - Algorithms (t=19, l=2)
      Algorithm: Shortest Path First (SPF) (0)
      Algorithm: Strict Shortest Path First (SPF) (1)
  ⌄ Node Maximum SID Depth (t=23, l=2)
      MSD Type: Base MPLS Imposition (1)
      MSD Value: 10
```

Container TLV

Sub-TLV 2
SR-MPLS with IPv4 control plane AF
Range is SRGB size (labels that can be allocated)

Sub-TLV 1
The starting label, so SR global block is 16000-23999 (typical one)

Sub-TLV 19
Algorithm in use

Sub-TLV 23
How many I can push (ref. label stack)

Capture – IP reachability

- IP reachability

```
Ext. IP Reachability: 10.10.10.1/32
  Metric: 0
  0... .... = Distribution: Up
  .1.. .... = Sub-TLV: Yes
  ..10 0000 = Prefix Length: 32
  IPv4 prefix: 10.10.10.1
  SubCLV Length: 11
 subTLV: Prefix-SID (c=3, l=6)
   Code: Prefix-SID (3)
   Length: 6
   Flags: 0x40, Node-SID
     0... .... = Re-advertisement: Not set
     .1.. .... = Node-SID: Set
     ..0. .... = no-PHP: Not set
     ...0 .... = Explicit-Null: Not set
     .... 0... = Value: Not set
     .... .0.. = Local: Not set
   Algorithm: Shortest Path First (SPF) (0)
   SID/Label/Index: 0x0000000a
```

'Normal' ISIS route

Represent Node-SID (identify a device in the topology, generally represented by a loopback)

Index is 0x0000000a = 10, hence label derived is 16010. (SRGB base + SID index)

Labels are derived and not signaled with SR contrary to other methods.

Capture – IS Reachability

- Extend IS reachability

```
˅ Extended IS reachability (t=22, l=68)
    Type: 22
    Length: 68
  ˅ IS Neighbor: 0000.0000.0004.00
      IS neighbor ID: 0000.0000.0004.00
      Metric: 50
      SubCLV Length: 23
    › subTLV: IPv4 interface address (c=6, l=4)
    › subTLV: IPv4 neighbor address (c=8, l=4)
    › subTLV: Link Maximum SID Depth (c=15, l=2)
    ˅ subTLV: Adj-SID (c=31, l=5)
        Code: Adj-SID (31)
        Length: 5
      › Flags: 0x30, Value, Local Significance
        Weight: 0x00
        .... 0000 0101 1101 1100 0001 = SID/Label/Index: 24001
  › IS Neighbor: 0000.0000.0003.00
```

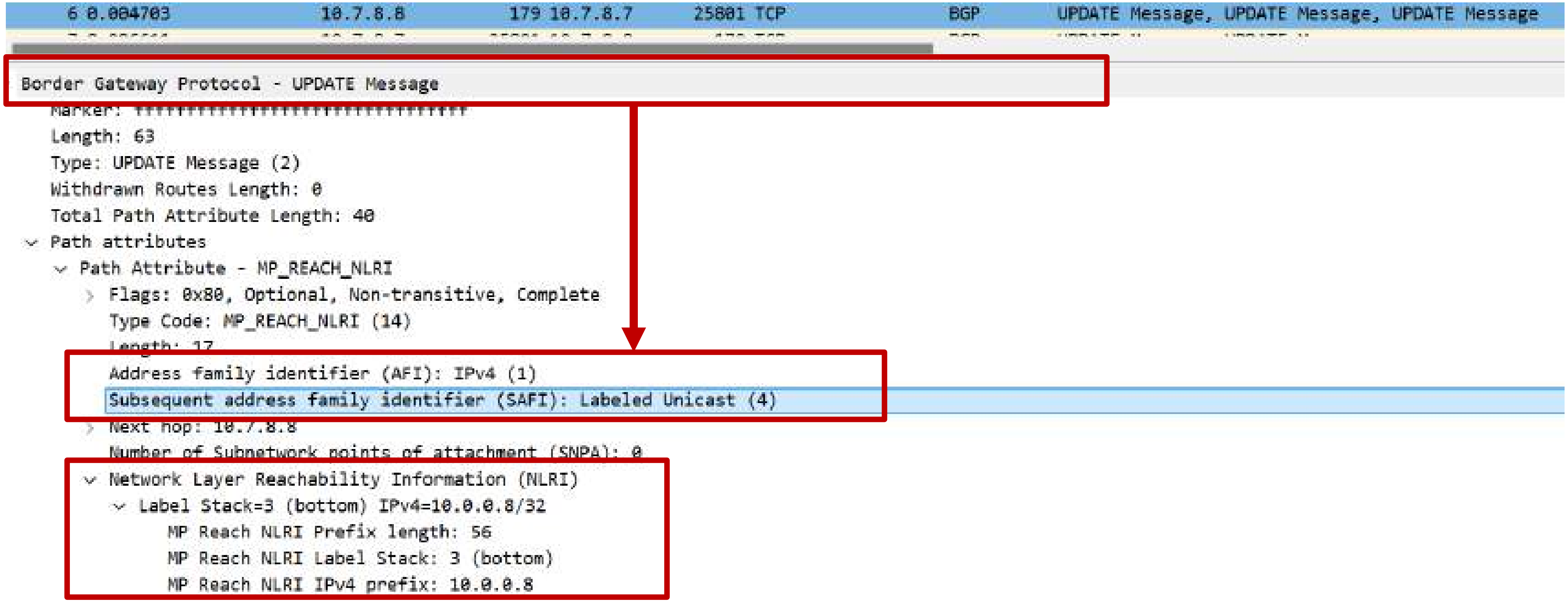
Link between routers

Adjacency SID, advertise SID associated with specific link

The significance is local only, meaningful for this router only.

Here the label value is directly included (explicit) and it is taken from SRLB starting at 24000.

- BGP IPv4 Labeled Unicast (AFI1/SAFI4)



https://web.archive.org/web/20230223022751/http://njrusmc.net/jobaid/bgp_pcap.zip

BGP Labeled Unicast

- With BGP-LU, you can exchange label between BGP speaking routers, without the need for the IGP help
  - it can be used to extend an MPLS network
    - Internally over different IGP islands/areas (OSPF/IS-IS),
    - Externally to another BGP AS (Inter-AS MPLS scenarios)
  - The label information will be attached to the NLRI
  - Use cases:
    - unified MPLS (merging multiple IGP islands),
    - inter-AS MPLS scenarios…
  - Support IPv6 as well (AFI2/SAFI4)

Quick summary until

- So, what we have seen until now… ☺

Layer 2 extension over MPLS

- As said, it is possible to carry frames with MPLS **(L2VPN)**

Outer frame (transport)

```
> Frame 22: Packet, 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface -, id 0
> Ethernet II, Src: aa:bb:cc:00:02:10, Dst: aa:bb:cc:00:01:10
> MultiProtocol Label Switching Header, Label: 1018, Exp: 0, S: 1, TTL: 252
> PW Ethernet Control Word
> Ethernet II, Src: aa:bb:cc:00:0a:20, Dst: aa:bb:cc:00:09:20
> Internet Protocol Version 4, Src: 172.16.90.10, Dst: 172.16.90.9
> Internet Control Message Protocol
```

Inner frame

https://web.archive.org/web/20230223022751/http://njrusmc.net/jobaid/mpls_pcap.zip

Layer operator

- The **layer operator (#)** can be used to match something against different occurrence of the field at different layer in case of multiple encapsulation

```
eth.addr#2==aa:bb:cc:00:09:20
```

lo.	Time	Source	SrcPort	Destination	DstPo	Protocol	MPLS TTL	Protocol
14	25.810017	172.16.90.…		172.16.90.9		ICMP	252	ICMP
15	25.810530	172.16.90.9		172.16.90.10		ICMP	255,255	ICMP
16	25.812017	172.16.90.…		172.16.90.9		ICMP	252	ICMP
17	25.812524	172.16.90.9		172.16.90.10		ICMP	255,255	ICMP
18	25.814013	172.16.90.…		172.16.90.9		ICMP	252	ICMP
19	25.814490	172.16.90.9		172.16.90.10		ICMP	255,255	ICMP
20	25.815912	172.16.90.…		172.16.90.9		ICMP	252	ICMP
21	25.816407	172.16.90.9		172.16.90.10		ICMP	255,255	ICMP
22	25.817858	172.16.90.…		172.16.90.9		ICMP	252	ICMP

```
Frame 22: Packet, 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface -, id 0
Ethernet II, Src: aa:bb:cc:00:02:10, Dst: aa:bb:cc:00:01:10
MultiProtocol Label Switching Header, Label: 1018, Exp: 0, S: 1, TTL: 252
PW Ethernet Control Word
Ethernet II, Src: aa:bb:cc:00:0a:20, Dst: aa:bb:cc:00:09:20
Internet Protocol Version 4, Src: 172.16.90.10, Dst: 172.16.90.9
Internet Control Message Protocol
```

https://web.archive.org/web/20230223022751/http://njrusmc.net/jobaid/mpls_pcap.zip

Capture

- What can you tell about this?

N	Time	Source	SrcPort	Destination	DstPo	Protocol	MPLS TTL	Protocol	Info
429	357.111252524	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
430	357.114027557	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
431	357.114882931	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
432	357.116943444	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
433	357.117564379	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
434	357.119299462	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
435	357.120293907	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
436	357.122001528	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
437	357.122681082	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
438	357.124433839	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
582	475.759615024	1.1.1.1	3503	127.0.0.1	3503	UDP	255	MPLS ECHO	MPLS Echo Request
583	475.764924770	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply

```
udp.port == 3503
```

Capture

- What can you tell about this?

No.	Time	Source	SrcPort	Destination	DstPo	Protocol	MPLS TTL	Protocol	Info
625	507.558757732	1.1.1.1	3503	127.0.0.1	3503	UDP	1	MPLS ECHO	MPLS Echo Request
626	507.565593840	10.1.11.11	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
627	507.571727781	1.1.1.1	3503	127.0.0.1	3503	UDP	2	MPLS ECHO	MPLS Echo Request
629	509.422683943	1.1.1.1	3503	127.0.0.1	3503	UDP	3	MPLS ECHO	MPLS Echo Request
630	509.427579862	10.44.4.4	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
631	509.428961753	1.1.1.1	3503	127.0.0.1	3503	UDP	4	MPLS ECHO	MPLS Echo Request
636	511.421736702	1.1.1.1	3503	127.0.0.1	3503	UDP	5	MPLS ECHO	MPLS Echo Request
638	511.430956700	10.22.33.33	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply
639	511.432112517	1.1.1.1	3503	127.0.0.1	3503	UDP	6	MPLS ECHO	MPLS Echo Request
640	511.436788765	10.33.3.3	3503	1.1.1.1	3503	UDP		MPLS ECHO	MPLS Echo Reply

```
udp.port == 3503
```

MPLS OAM

- **MPLS OAM** (Operations, Administration, Maintenance) is used to detect operational failures but can also be leveraged for performance measurements, accounting, and service monitoring (RFC4377, RFC4379, RFC6669, and more)
  - Identify control and data plane defects,
  - Identify LSP defects,
  - Get path information,
  - Get SLA information,

MPLS LSP ping & MPLS LSP traceroute utilities are generally found on most network operating system supporting MPLS.

MPLS OAM

SharkFest'25 EUROPE

- **MPLS ping and traceroute utilities** rely on the same OAM messages

```
> Frame 586: Packet, 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface eth0, id 0
> Ethernet II, Src: 50:02:00:15:00:02 (50:02:00:15:00:02), Dst: 50:02:00:0f:00:06 (50:02:00:0f:00:06)
> MultiProtocol Label Switching Header, Label: 24001, Exp: 0, S: 1, TTL: 255
v Internet Protocol Version 4, Src: 1.1.1.1, Dst: 127.0.0.1
    0100 .... = Version: 4
    .... 0110 = Header Length: 24 bytes (6)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 96
    Identification: 0x641f (25631)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
    Protocol: UDP (17)
    Header Checksum: 0xff66 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 1.1.1.1
    Destination Address: 127.0.0.1
  v Options: (4 bytes), Router Alert
    > IP Option - Router Alert (4 bytes): Router shall examine packet (0)
```

MPLS header (TTL set to 255 in case of ping)

IP header with destination address set to 127.0.0.x

Options Router Alert set to punt the packet to control plane

- **MPLS OAM** data inside **UDP/3503** datagram

```
> User Datagram Protocol, Src Port: 3503, Dst Port: 3503
v Multiprotocol Label Switching Echo
    Version: 1
  > Global Flags: 0x0000
    Message Type: MPLS Echo Request (1)
    Reply Mode: Reply via an IPv4/IPv6 UDP packet (2)
    Return Code: No return code (0)
    Return Subcode: 0
    Sender's Handle: 0xf79e33f1
    Sequence Number: 3
    Timestamp Sent: Nov  4, 2025 21:07:49.711999999 UTC
    Timestamp Received: Jan  1, 1970 00:00:00.000000000 UTC
  v Vendor Private
      Type: Vendor Private (64512)
      Length: 12
      Vendor Id: ciscoSystems (9)
      Value: 0001000400000004
  v Target FEC Stack
      Type: Target FEC Stack (1)
      Length: 12
    > FEC Element 1: Generic IPv4 prefix
```

UDP port 3503 reserved for MPLS Echo

MPLS OAM payload carrying all the information that will be interpreted

MPLS OAM

- **MPLS traceroute** will just send the same MPLS Echo Request, but with **incremented MPLS TTL values**

625	507.558757732	1.1.1.1	3503 127.0.0.1	3503 UDP	1	MPLS ECHO MPLS Echo Request
626	507.565593840	10.1.11.11	3503 1.1.1.1	3503 UDP		MPLS ECHO MPLS Echo Reply
627	507.571727781	1.1.1.1	3503 127.0.0.1	3503 UDP	2	MPLS ECHO MPLS Echo Request
629	509.422683943	1.1.1.1	3503 127.0.0.1	3503 UDP	3	MPLS ECHO MPLS Echo Request
630	509.427579862	10.44.4.4	3503 1.1.1.1	3503 UDP		MPLS ECHO MPLS Echo Reply
631	509.428961753	1.1.1.1	3503 127.0.0.1	3503 UDP	4	MPLS ECHO MPLS Echo Request
636	511.421736702	1.1.1.1	3503 127.0.0.1	3503 UDP	5	MPLS ECHO MPLS Echo Request
638	511.430956700	10.22.33.33	3503 1.1.1.1	3503 UDP		MPLS ECHO MPLS Echo Reply
639	511.432112517	1.1.1.1	3503 127.0.0.1	3503 UDP	6	MPLS ECHO MPLS Echo Request
640	511.436788765	10.33.3.3	3503 1.1.1.1	3503 UDP		MPLS ECHO MPLS Echo Reply

More here: https://www.cisco.com/c/en/us/support/docs/multiprotocol-label-switching-mpls/mpls/200510-Trace-route-in-MPLS-network.html

About TTL

- Speaking about TTL,
  - Routers can be configured in two ways regarding TTL
  - RFC3443
    - Either the IP TTL is copied inside the MPLS TTL at the ingress point, and the MPLS TTL (that has been decremented inside the carrier's network) is copied back at the egress point the IP TTL (sometimes referred as uniform mode)
    - Either the IP TTL is left untouched, the label is pushed with TTL set to 255 and get decremented inside the carrier backbone. This later option will hide part of the service provider network (sometimes referred as pipe model)

Common issues

- Some issues
  - Label distribution problems
    - IGP/LDP not sync'ed (link in IGP up but no LDP session),
    - Label exchange issues (BGP,IGP…),
    - Configuration and consistency issues across the network,
    - Defects in network operating system daemon/diagnosing interop issues,
  - Data-plane related issues
    - Label missing in LFIB, label withdrawn but not reinstalled (bug…)
    - Inconsistent platform support for implicit-null / explicit-null
    - Missing route recursion resolution
    - Defect in ASIC programming (bug or hardware limitation)
    - **MTU…** ☺

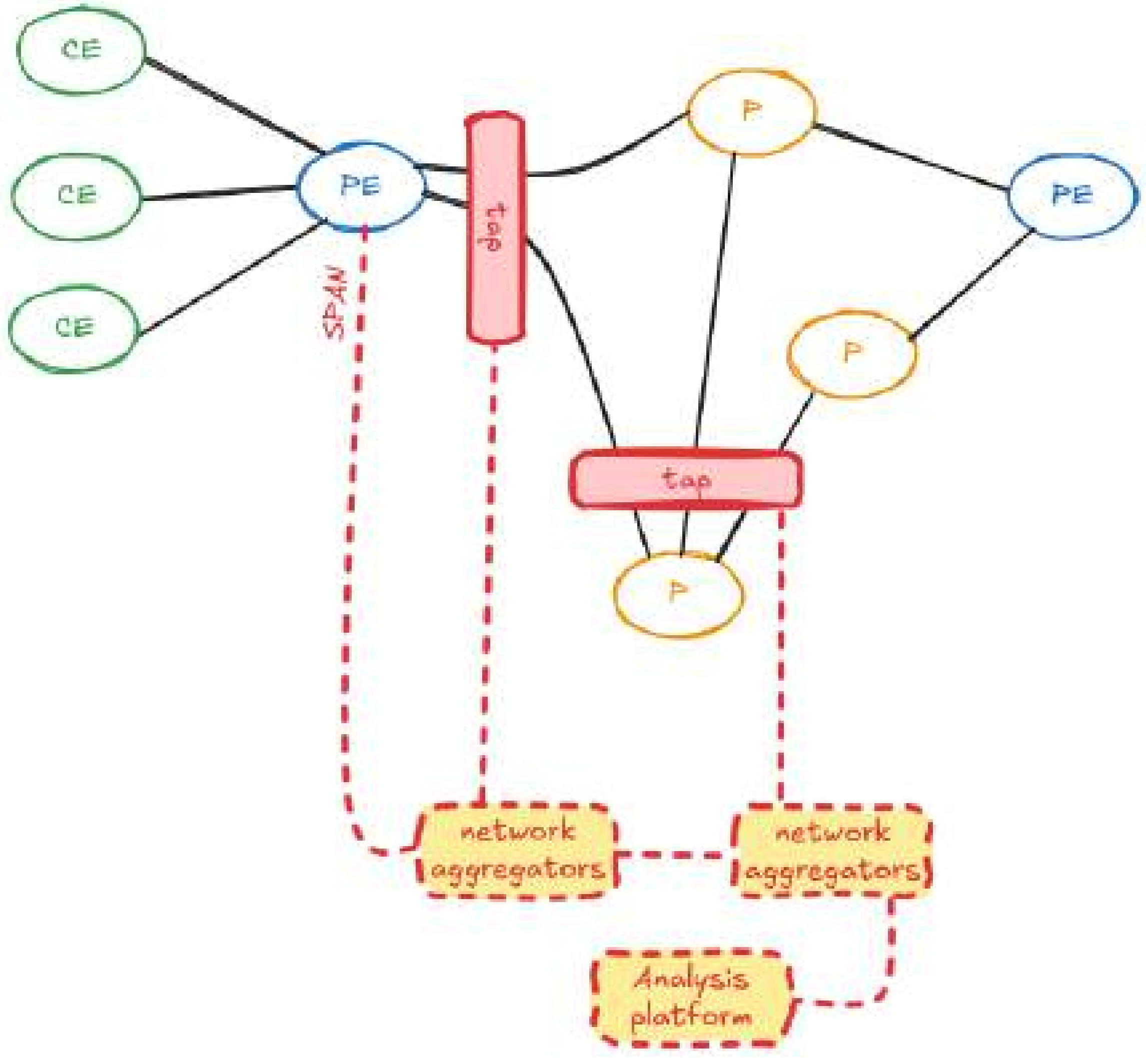
Capture strategies

- As always, capturing packets in large-scale, heavy traffic environments presents a unique set of challenges
  - Variety of links (type of media),
  - Variety of bandwidths (10G, 40G, 100G, more…),
  - Distributed architectures with many paths,
  - Potential impact on production systems,
  - Variety of capture points,
  - Time synchronization,

Credits: https://www.profitap.com/fiber-taps/

Capture strategies

- In **dense environments**, a **specific network** might be built to handle packet capture traffic, and its distribution to analysis tools
  - This network can **aggregate multiple sources** like on-the-box *SPAN or network taps / packet brokers
  - **Capture filters** may be leveraged early on to **reduce the volume** of **data captured** (BPF-based filters, for example)
  - **NIC card bandwidth** inside the capture infrastructure needs to be **consistent**



Capture strategies

- In the end, why do we do that?
  - **Learning and experiments** (NOS configuration and behaviors, multi-vendor verifications…)
  - Traffic **analytics**, **application performance/troubleshooting** to some extent,
  - IT **security** and **incident response** (sending traffic for analysis to IDS/IPS,…)
  - **Forensic, lawful interception** may be,

Depending on the objectives, the toolset in place will be adjusted.

Capture strategies

- Common issues that need to be solved
  - Traffic volume, **filters** need to be applied,
  - Privacy/confidentiality and storage capacity, **packet payloads may need to be chopped**,
  - Timestamps may confuse **performance analysis**; some packet brokers can perform **timestamping**,
  - Packet encapsulation needs to be removed so the end system get the information it expects (**header stripping**),
  - Encryption,

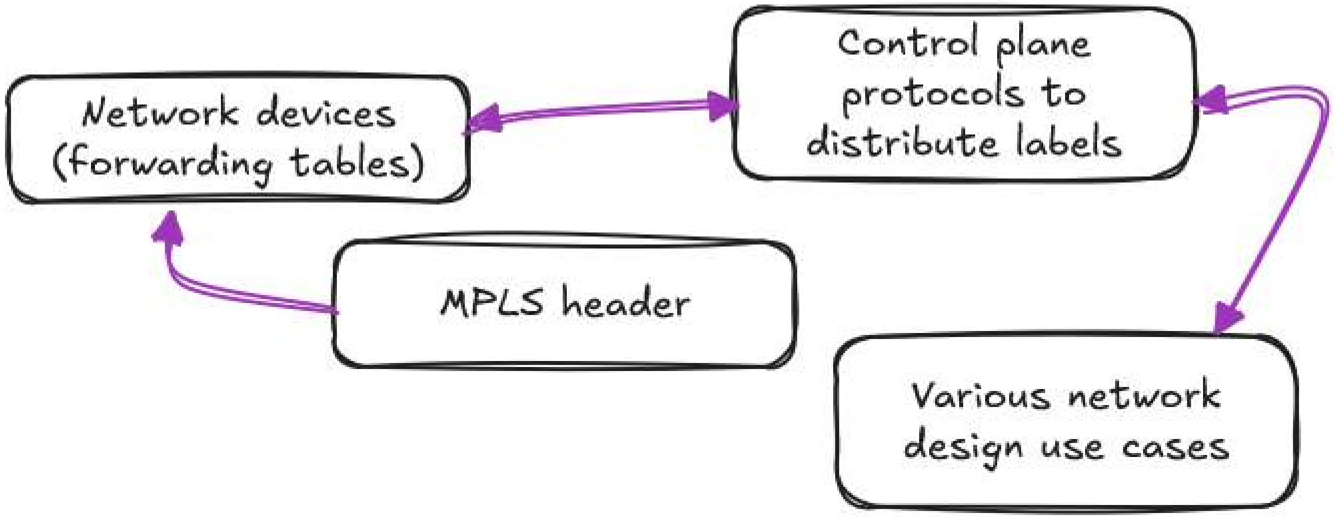
Wireshark capture filters

- Capture filters in wireshark
  - Set before capturing, allow for precise capturing, removing potential unnecessary traffic,
  - Cannot be changed once capture is started,
  - Be careful, some important may be omitted…
  - Capture filters leverage libpcap pcap-filters (tcpdump syntax for those familiar), they are different than wireshark display filters

  - https://wiki.wireshark.org/CaptureFilters
  - https://sharkfest.wireshark.org/retrospective/sfus/presentations16/13.pdf

Key take-aways

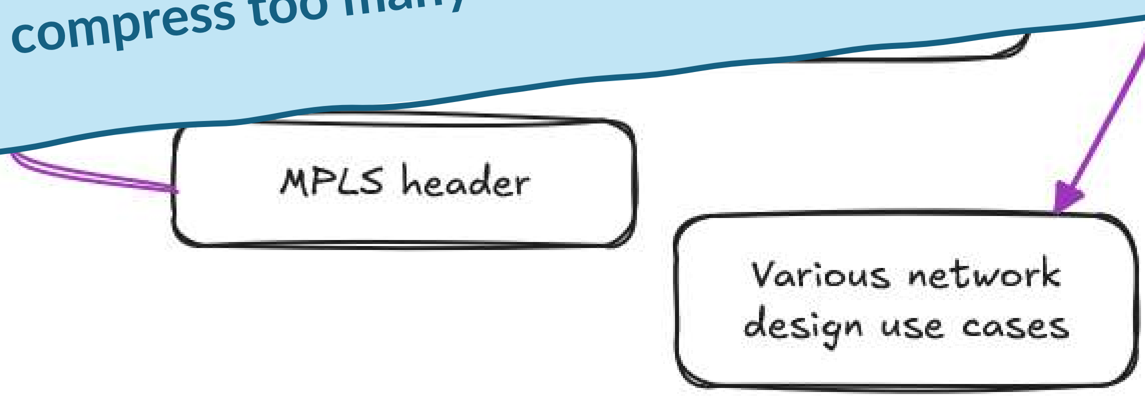
- You learned,
    - The principle of label-based forwarding,
    - The MPLS header,
    - New acronyms,
    - How the label information is exchanged between routers,
    - What are MPLS VPNs and the label stack
    - Some capture insights, and tips hopefully



Key take-aways

- You learned,
  - The principle of label-based forwarding,
  - The MPLS header,
  - New acronyms,
  - How the label information is exchanged between routers,
  - What are MPLS VPNs and the label stack
  - Some capture insights, and ti

Pablo's Note:
Don't try to compress too many MPLS topics in 50 minutes



Questions?

We kindly ask for your feedback!

- Thank you very much for attending
- Your feedback is welcomed!

https://conference.wireshark.org/sharkfest-25-europe-2025/talk/K3QZUJ/feedback/

#sf25eu